

MSC.ADAMS/CAR 教程

目录表

0.0 欢迎参加 MSC.ADAMS/CAR培训

MSC.ADAMS软件的简史

MSC.Software公司简介

如何得到帮助

有关MSC.ADAMS的基本术语简介

第一章 ADAMS/CAR 简介

为什么要使用 ADAMS/Car ?

用户模式

数据库结构 — 目录的谱系结构

数据库转换升级

数据库中文件的转换升级

存储文件：工作目录与数据库

配置文件

第二章 练习 1: 打开并运行一个装配 基本概念

数据的谱系关系

试验台

主要角色与次要角色

命名规则

第三章 练习 2: 模板 VS.子系统 生成并调整子系统

生成子系统

修改子系统

调整硬点

调整参数变量

调整质量特性

将刚性体替换为弹性体

目录表(续)

第 4 章

调整弹簧和阻尼器

对子系统其它可用的调整

实体(Instance)定义的替换

练习 3: 生成并调整子系统

曲线编辑器和特性文件编辑器

利用曲线编辑器和特性文件编辑器编制特性文件

特性文件种类

生成特性文件

特性文件编辑器

转换 TeimOrbit 格式文件为 XML 格式文件

修改现有的特性文件

绘图 vs. 数表

练习 4: 使用曲线编辑器修改弹簧特性

第 5 章

输出配置文件

绘图配置文件

生成绘图配置文件

利用绘图配置文件生成绘图

绘图配置文件的格式

绘图配置文件的例子

练习 5 : 生成绘图配置文件

第 6 章

生成并仿真悬架系统

生成悬架装配

二分之一车辆分析

悬架参数

生成载荷工况

警告信息

分析所对应的结果文件

练习 6 : 运行悬架分析

目录表(续)

第 7 章

生成并仿真整车模型

生成整车装配

移动子系统的装配位置

更新子系统和装配

使子系统和装配同步

增加/减少和激活/失效子系统

调整质量特性

整车分析

整车分析事件生成器

练习 7: 进行整车分析

第 8 章

驾驶机器

标准驾驶员界面 (SDI) 和驾驶机器

为什么使用 SDI?

数据流程

使用驾驶机器

限制

生成 SDI 的输入

驾驶员控制文件(Driver Control Files)

驾驶员控制文件(.dcf)的例子

生成 .dcf 和 .dcd 文件

驾驶员控制文件(.dcf)的结构

指定结束条件

可扩展的结束和退出条件

指定属性

驾驶员控制数据文件(Driver Control Data Files)

驾驶员控制数据文件(Driver Control Data Files)的构造体系

生成 .dcd 文件

练习 8: 编辑 .DCF 和 .DCD 文件

目录表(续)

第 9 章

参数化

在 ADAMS/Car 中的参数化

生成硬点

生成构造坐标系

位置参数化

方向参数化

第 10 章

建立模板

模板综述

模板的拓扑关系

文件的构成

建立一个新的模板

部件的类型

刚体和弹性体 (部件)

几何外形

连接方式 (约束和衬套)

弹簧

阻尼器

止挡和反向止挡

前束角(Toe)/外倾角(Camber)和悬挂参数数组

可调载荷

一些建议

练习 9: 模板建模入门

第 11 章

信息交换器

信息交换器的种类

信息交换器的分类

信息交换器的几何对称性

信息交换器的角色

信息交换器的命名

目录表(续)

在装配过程中信息交换器的匹配

与试验台之间信息交换器的匹配

练习 10: 取得信息交换器的信息

第 12 章

使用弹性体

弹性部件综述

使用弹性部件的一些限制

得到弹性部件

模态叠加

可视化属性

有关约束和驱动

约束连接的限制

练习 11: 弹性部件入门练习

第 13 章

测试请求

测试请求激活的切换

第 14 章

轮胎

轮胎综述

ADAMS/Tire 模块

Tire 模型

路面类型

2D 路面的例子 (RAMP)

使用 3D 等效体积模型

3D 路面的例子

3D 光滑路面

路面数据文件的实例

使用 3D 光滑路面

如何使用 ADAMS/Tire

练习 12: 建立一个轮胎的模板

目录表(续)

第 15 章

模板的调研

调研模板

理解模板

关于 Database Navigator

第 16 章

练习 13: 调研并完成模板

其它应用

即插即用功能

ADAMS/Car Ride——平顺性分析模块

ADAMS/Vibration——振动分析模块

ADAMS/Durability——耐久性分析模块

ADAMS/Driveline——传动系模块

ADAMS/Linear——线性化模块

ADAMS/Controls——控制模块

ADAMS/Insight——试验分析模块

练习 14: 在 CAR 中使用 LINEAR

练习 15: 整车装配

附录 A

分析的例子

分析类型

搜集有关模型的数据

悬架的包络空间分析

悬架的运动学分析

悬架的柔顺性分析

静载耐久性分析

动载耐久性分析

前悬架分析

整车的设计和分析

附录 B

四轮台柱式垂向激励试验台

附录 C

ADAMS/CAR 中的文件

0.0 简介

欢迎参加 MSC.ADAMS/CAR培训

- 欢迎参加 **MSC.ADAMS/CAR**培训。在本教程中，你将学习如何使用 **ADAMS/Car** 建立并分析悬架和车辆的装配模型。

欢迎参加 MSC.ADAMS/CAR培训

- 本章内容：
 - ◆ MSC.ADAMS软件的简史
 - ◆ MSC.Software公司简介
 - ◆ 如何得到帮助
 - ◆ 有关MSC.ADAMS的基本术语简介

MSC.ADAMS的发展简史

- ADAMS (Automatic Dynamic Analysis of Mechanical Systems)：机械系统自动化动力学仿真软件。
- 1977年成立，发展史可以追溯到28 年前。
- Mechanical Dynamics Incorporated (MDI) 由美国密西根大学的ADAMS 代码开发研究人员发起成立，位于美国密西根州的Ann Arbor，2002年MSC.Software 以1.2亿美金收购了MDI公司。
- 系统级运动仿真。
- 机构大位移运动分析代码。
- 最开始只有 ADAMS/Solver，用来解算非线性的方程组。你需要以文本方式建立模型提交给 ADAMS/Solver 进行求解。
- 在90年代初，ADAMS/View 发布了，你可以在统一的环境下建立模型、仿真模型和检查结果。
- 现在，已经发布了一些基于不同行业的产品，比如：ADAMS/Car、ADAMS/Rail 和 ADAMS/Engine等。

MSC.SOFTWARE公司简介

- 有关 **MSC.Software** 公司的系列产品，请查阅：
 - ◆ <http://www.mscsoftware.com/products/products.cfm>
- 有关 **MSC.ADAMS** 的相关产品，请查阅：
 - ◆ http://www.mscsoftware.com/products/products_detail.cfm?PI=413
- 有关 **MSC.ADAMS** 各区的培训，请查阅：
 - ◆ <http://www.engineering-e.com/training/>
 - ◆ 或你区的办事处
- 有关 **MSC.ADAMS** 解算器验证问题，请查阅：
 - ◆ <http://support.adams.com/kb/faq.asp?ID=kb9587.dasp>

如何得到帮助

■ 在线帮助



◆ 要使用在线帮助：

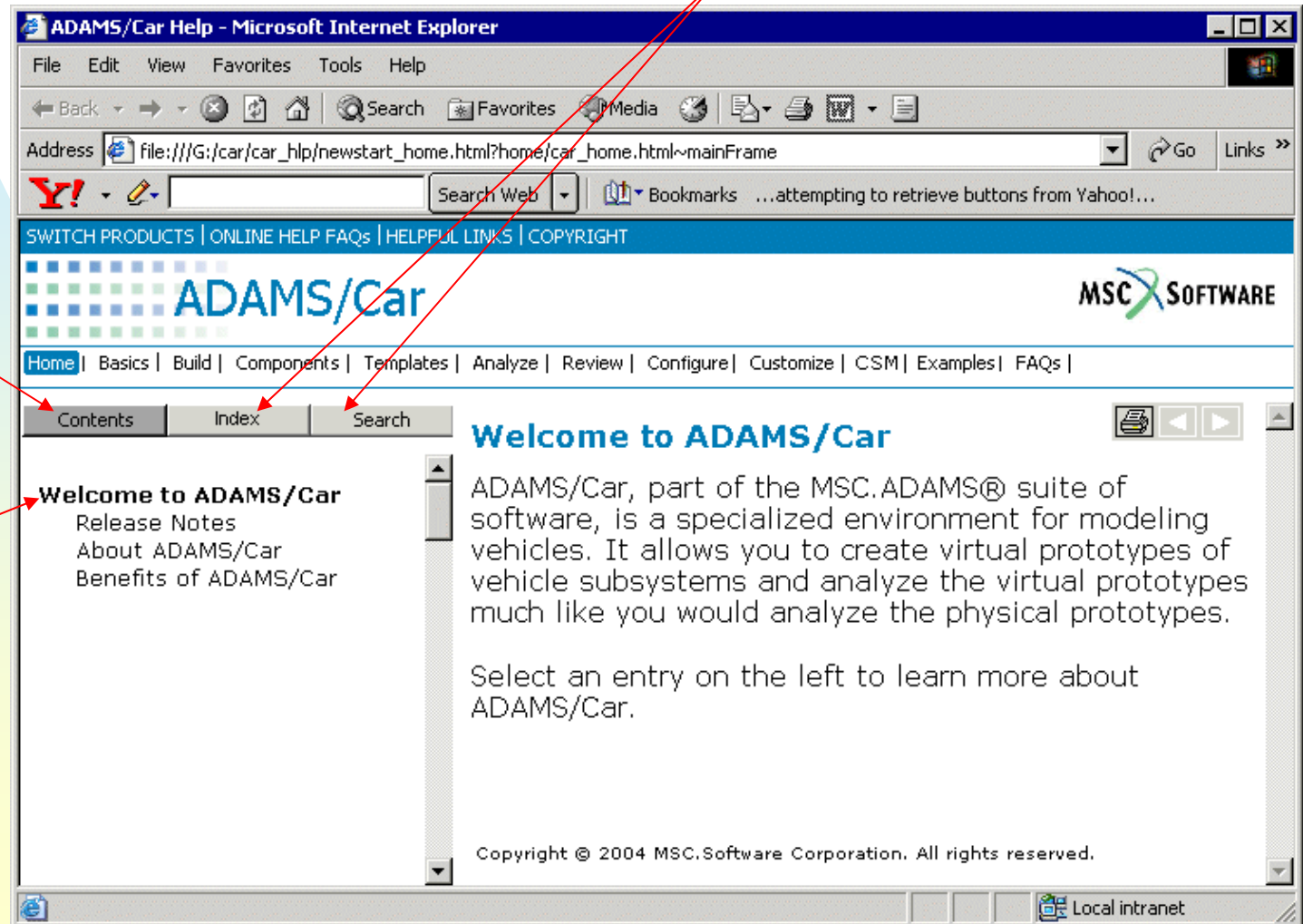
- 从界面中的 **Help** 菜单，选择 **ADAMS/View Help**，可以显示 ADAMS/View 在线帮助的根本文件。
 - 在 ADAMS/View 下任意一个对话框中，按下 **F1** 键显示有关该对话框的帮助。
- ◆ 一旦在线帮助的对话框打开后，你可以通过**浏览**或**索引**或**搜索**的方式找到你所需要的帮助条目。

如何得到帮助

ADAMS/Car 全部帮助
的索引/搜索

被选内容

浏览条目的
目录表



如何得到帮助

■ 技术支持

- ◆ 欲找到各地区技术支持，可浏览：
<http://www.mscsoftware.com/support/contacts/index.cfm>
- ◆ 欲查找服务协议，可浏览：
http://www.mscsoftware.com/support/prod_support/adams/ADM_02ZZZLT_T_SERL_HJ_R6.pdf

■ 知识库（KB）

- ◆ 浏览：<http://support.adams.com/kb>
- ◆ 漫游相关网页，请浏览：
<http://www.adams.com/news/newsletter/vol3/kbtour.htm>

■ 咨询服务

- ◆ <http://www.mscsoftware.com/services/esg/>

如何得到帮助

- **MSC VPD (Virtual Product Development) 社区**
 - ◆ **MSC.Software** 用户欲加入社区，可进入：
<http://forums.mscsoftware.com>。
 - ◆ 选择 **MSC.ADAMS** 浏览有关 **MSC.ADAMS** 的讨论。
 - ◆ 选择 **MSC News** 浏览产品信息及公司新闻事件。

有关MSC.ADAMS的基本术语简介

- ADAMS/Solver
 - ◆ 仿真发动机。
- ADAMS/Solver dataset (*.adm)
 - ◆ 一个提交 ADAMS/Solver的ASCII格式模型文件，包括定义模型的卡片和函数两部分。
- ADAMS/Solver command (*.acf)
 - ◆ 包含控制ADAMS/Solver 解算模型所需要的命令。

有关MSC.ADAMS的基本术语简介

■ ADAMS/Solver 输出文件

- ◆ **Graphics** (*.gra) – 包含有关模型图形方面的信息文件。
- ◆ **Request** (*.req) – 包含用户指定结果的输出文件。
- ◆ **Results** (*.res) – 包含模型中每个对象的状态变量（state-like）结果。通常而言，此文件非常大，按照缺省方式，ADAMS/Car一般不输出此文件；但你可以强制要求ADAMS/Car输出此文件。对于包含弹性体的模型，在动画时，需要此文件。
- ◆ **Message** (*.msg) – 有关 ADAMS/Solver 仿真过程的信息文件。结合DEBUG/EPRINT 使用，有助于你调试模型。
- ◆ **Output** (*.out) – 包含初始条件和输出请求（request），内容跟输出选项有关。通常是数表格式并且可读（例如：使用 LINEAR命令的数表输出）。

第一章 ADAMS/CAR 简介

- 本章主要介绍使用ADAMS/Car 的优势所在以及一些基本的文件的组织结构关系。

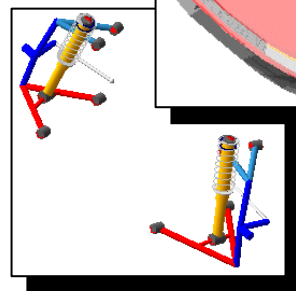
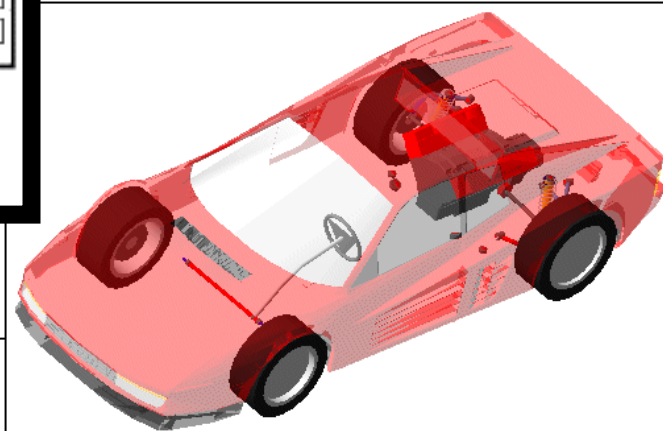
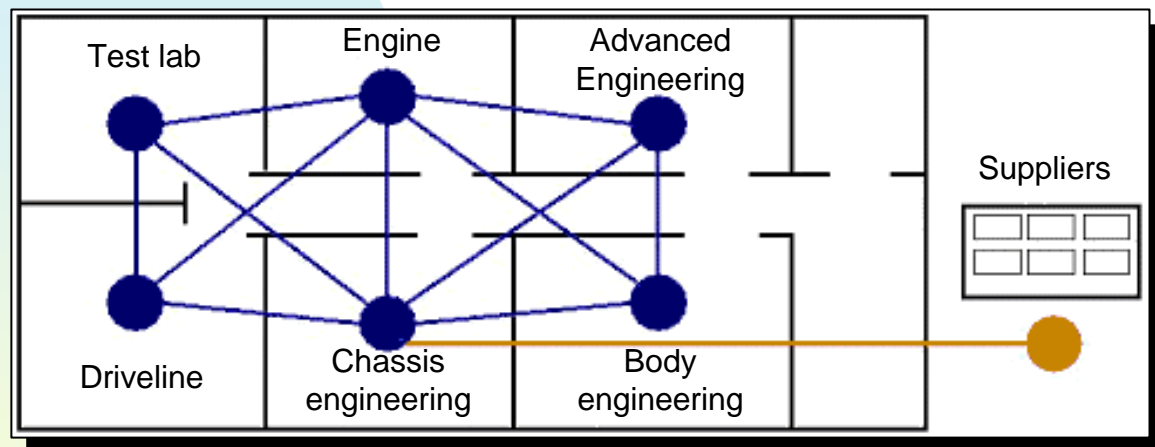
ADAMS/CAR 简介

■ 本章内容：

- ◆ 为什么要使用 ADAMS/Car？
- ◆ 用户模式
- ◆ 数据库结构 — 目录的谱系结构
- ◆ 数据库转换升级
- ◆ 数据库中文件的转换升级
- ◆ 存储文件：工作目录与数据库
- ◆ 配置文件

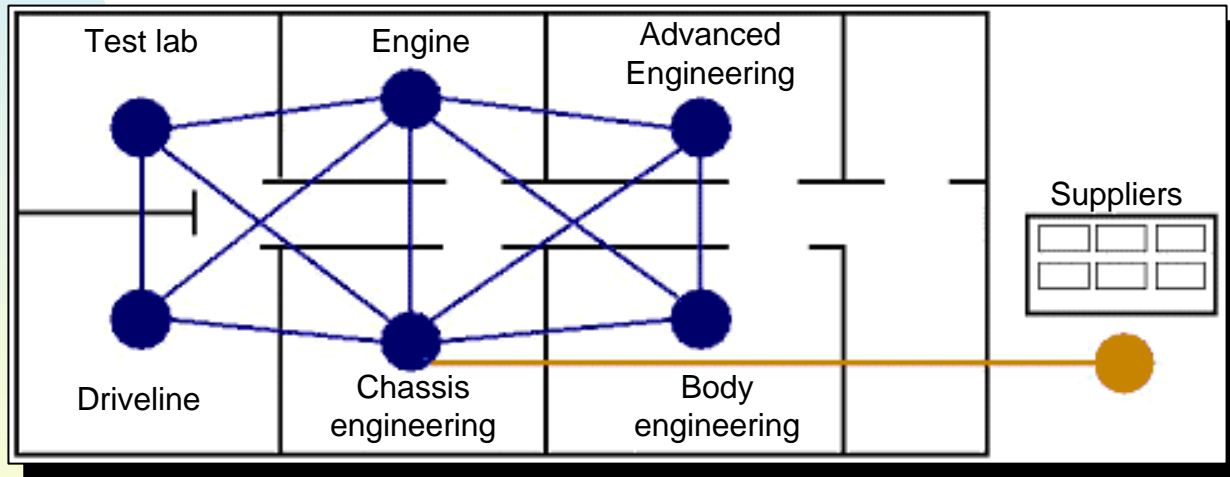
为什么要使用 ADAMS/Car ?

- 通过共享模型和数据在各部门及产品供应商间提供桥梁
- 方便地进行子系统迅速更换
- 模板方式建模



为什么要使用 ADAMS/Car ?

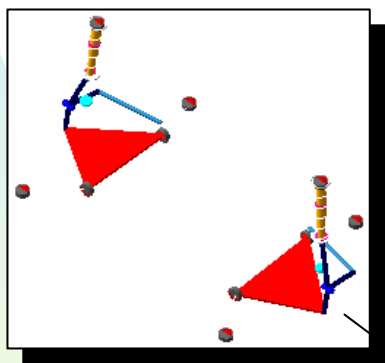
- 通过共享模型和数据在各部门及产品供应商间提供桥梁
 - ◆ 不同的部门可以在统一的数据库下工作，最大限度的减少数据损失。



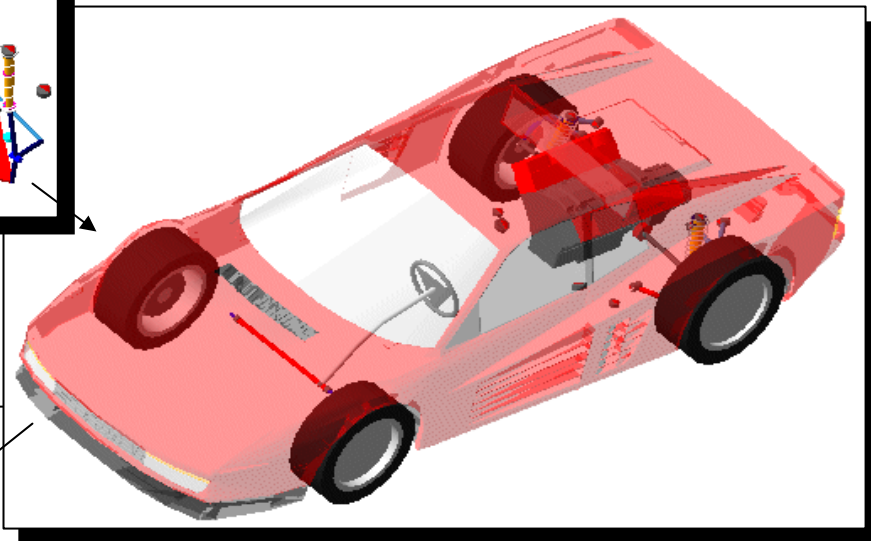
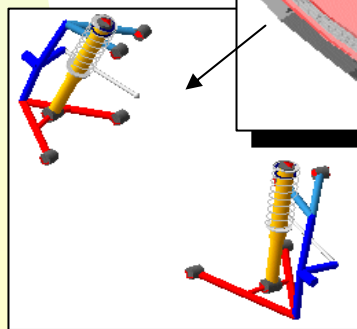
为什么要使用 ADAMS/Car ？

- 方便地进行子系统迅速更换
 - 你可以很容易的替换一个子系统，而无须改变车辆模型中的其它部件。

MacPherson



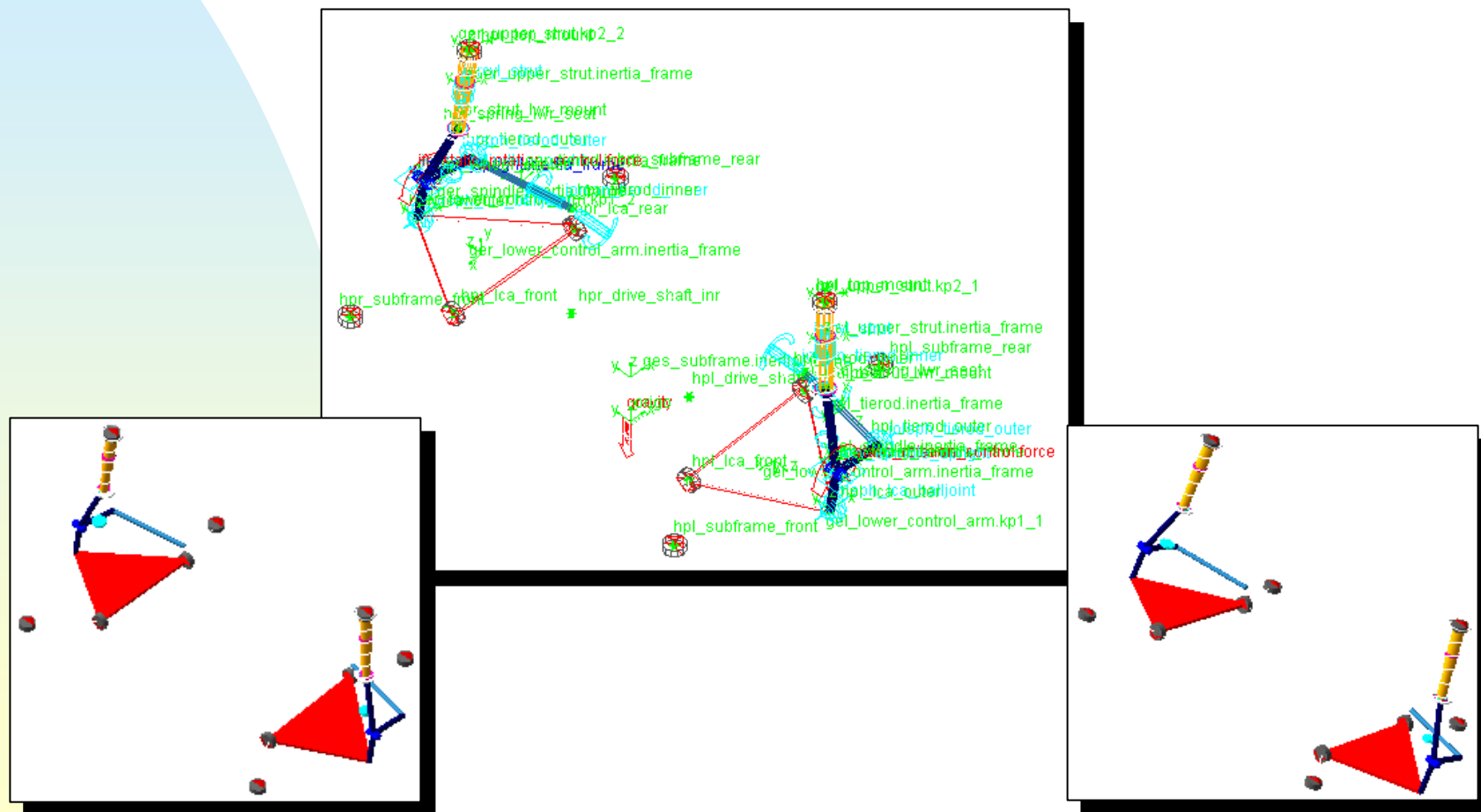
SLA



为什么要使用 ADAMS/Car ?

模板方式建模

- ◆ 允许你为多个车辆模型只制作一个子系统的模板。



用户模式

- ❑ 在 ADAMS/Car 的配置文件 (acar.cfg 和 .acar.cfg) 内，ADAMS/Car 有两种模式可以指定，一种是标准用户模式，另一种是专家用户模式。

- 专家用户模式：

Expert user (Template Builder 和 Standard Interface)

- ◆ 通过 Template Builder，允许用户建立 ADAMS/Car 的模板。
- ◆ 适合于 MSC.ADAMS 有经验的用户
- ◆ 允许用户使用 MSC.ADAMS 所有的建模实体

- 标准用户模式

Standard user (Standard Interface only)

- ◆ 特别适合设计人员和试验工程师使用
- ◆ 使用 ADAMS/Car 中的数据库，很容易生成车辆的子系统和装配
- ◆ 按照汽车试验标准提供仿真环境

数据库结构 — 目录的谱系结构

- 一个数据库就是存储在硬盘上一系列的表（目录）。最顶层目录，其目录名称有扩展名.cdb，存储若干个表（目录），每个表为存储模型信息的地方。

- 数据库有三种类型：
 - ◆ 共享型(**Shared**) – 由MSC.Software公司提供的一些例子，对所有用户开放。
 - ◆ 个人型(**Private**) – 缺省用户工作空间，（由ADAMS/Car在用户的\$HOME目录下自动生成的数据库）。
 - ◆ 用户型(**User**) – 由User/site进行指定。



数据库结构 — 目录的谱系结构

- 数据库由个人目录下的.acar.cfg或公共目录下的acar.cfg定义。
- 数据库的数量没有限制。
- (通常情况下)每个项目应该分别设定一个单独的数据库。
- 每次你只能向一个数据库存储文件。



有关如何管理ADAMS/Car的数据库，请参见
ADAMS/Car的在线帮助手册。

数据库转换升级

- 从ADAMS/Car 2003起，允许在ADAMS/Car中ADAMS/Driveline和 ADAMS/Engine作为插件使用。这样可以通过增加详细的发动机和传动系模型提高你模型的逼真程度，也允许你建立更完整的功能化数字车辆。
- 由于插件的变化，有必要对ADAMS/Car、ADAMS/Engine和ADAMS/Driveline的共享数据库进行改名。在12.0版，ADAMS/Car、ADAMS/Driveline和ADAMS/Engine模块分别按照下列别名以及被存储在硬盘上名称：

数据库转换升级

Product:	Database alias:	Database name on disk:
ADAMS/Car	<shared>	shared_car_database.cdb
ADAMS/Driveline	<driveline>	driveline.cdb
ADAMS/Engine	<shared>	shared_engine_database.cdb

- 我们已经将所有的三个产品的共享数据库名称进行了标准化，如下表所示：

Product:	Database alias:	Database name on disk:
ADAMS/Car	<acar_shared>	shared_car_database.cdb
ADAMS/Driveline	<adriveline_shared>	driveline.cdb
ADAMS/Engine	<aengine_shared>	shared_engine_database.cdb

数据库转换升级

- 由于这些改变，你必须修改所有现存的用户型数据库以对应于新版的MSC.ADAMS的数据库。下面教授你如何使用MSC.ADAMS提供的一个python来修改用户型数据库。
- 要修改用户型数据库：
 - 注意：我们推荐你在进行每个数据库转换升级之前最好先将文件做备份。
 - ◆ 从菜单**Tools**下指向**Database Management**再指向**Version Upgrade**，然后选择**Shared Database Alias**。
 - ◆ 如果你使用未经转换的数据库CDB运行ADAMS/Car，缺省的假定，任何的 <shared> 均为 <acar_shared>，因此你可以直接使用现存的用户型数据库。
 - ◆ 如果你是功能化数字车辆的用户，你需要进行数据库的转换，以便不会出现ADAMS/Car的<shared>和ADAMS/Engine的<shared>数据库CDBs出现混淆的问题。

数据库中文件的转换升级

- 通常情况下，ADAMS/Car在打开模版或子系统时，自动地将模板转换为新版本。这一过程尚不是很费时间。如果你将所有的模版一次性地转换升级可以节省一些时间，胜过每次打开时再转换。你可以使用在工具Database下的Convert Files完成这一切的操作。
- 使用该对话框的功能需要一些经验，并且应该慎重使用，因为未经测试是否满足需要的质量矩阵。你可以使用该对话框自动完成数据库的升级，从一个版本，如：11.0，到新版本，如：12.0。

数据库中文件的转换升级

■ 要打开Database对话框下的Convert Files：

1. 从菜单**Tools**下，指向**Dialog Box**，然后再选择**Display**。
出现了Database Navigator对话框。

2. 找到 **dbbox_too_dat_con** 并双击鼠标左键打开。

下表为完成该对话框所需要的一些选项：

选项名称	进行下列操作：
数据库名称	选择你要转化到最新版本的数据库名称。
目标数据库名称	选择一个可写入的数据库作为更新数据库，如果不选的话，为缺省的可写数据库。
文件类别	选择你要转换文件的类型。
数据库信息	选择此项显示信息窗口，列出当前可以使用的数据库。

存储文件：工作目录与数据库

- ADAMS/Car将悬架或车辆模型分成若干子系统和，以便你能够更方便的管理这些数据。ADAMS/Car将这些构造部分存储在一个ADAMS/Car的数据库下。最终目标是生成 .adm 文件并得到分析结果，这些文件是存储在工作目录下的。
- 你可以设置组装模型的构造部分等的存储数据库，通过数据库管理工具 (Tools → Database Management) 完成。
- ADAMS/Car 存储你模型的 .adm、.acf 和结果文件在你的工作目录下，设置方法为：File → Select Directory。

配置文件

- 配置文件包含一些**ADAMS/Car**启动过程要正确初始化所需要读进来的一些必要信息：
 - ◆ User mode (expert versus standard)
 - ◆ Personal databases and tables
 - ◆ Default property files
 - ◆ Default writable database
 - ◆ Database search order
 - ◆ Orientation of global reference frame
 - ◆ Other preferences

配置文件

```
!-----!  
!***** ADAMS/Car Configuration File *****!  
!-----!  
! - List of personal environment variables  
!-----!  
ENVIRONMENT MDI_ACAR_USERMODE      expert  
ENVIRONMENT MDI_CAR_RENDER          sshaded  
ENVIRONMENT MDI_ACAR_VEHICLE_REAR   1, 0, 0  
ENVIRONMENT MDI_ACAR_VEHICLE_LEFT   0, -1, 0  
!  
!-----!  
! - List of personal database directories  
!-----!  
!                                     Database namePath of database  
!-----!  
DATABASE      private      D:\private.cdb  
DATABASE      dbase_1      D:\dbase_1.cdb  
DATABASE      dbase_2      D:\dbase_2.cdb  
DEFAULT_WRITE_DB  private  
!  
!-----!  
! - Desired database search order  
!-----!  
DATABASE_ORDER      private, dbase_1, dbase_2, shared  
!  
!-----!
```



有关配置文件更多的信息，请参见**ADAMS/Car**的在线帮助手册。

配置文件

■ 共享配置文件与个人配置文件

◆ ADAMS/Car使用两个配置文件：

- 共享配置文件(**acar.cfg**) – 为共享文件，用以设置缺省的参数(首先读入)。该文件在<adams_install>/acar目录下，(其中<adams_install>为你的 **MSC.ADAMS**安装目录)。
 - 个人配置文件(**.acar.cfg** – 注意在文件前面有一个点“.”) – 为个人文件，同样用以设置参数，但是优先于共享配置文件**acar.cfg** (再次读入)。个人配置文件在个人目录**HOME**下 (**HOME**为系统的环境变量)。
- ### ◆ 使用两个配置文件，用户可以按照缺省的参数使用软件，并且利用个人配置文件.acar.cfg实现其机器上的个性化设置。

配置文件

■ 存储配置文件

◆ 可以按照下列方式存储你的配置文件：

- 共享配置文件(**acar.cfg**) – 使用文本编辑工具手工打开并存储该文件。该文件位于<ampadams_install>/acar目录下(其中：<ampadams_install> 为**MSC.ADAMS**安装目录)。通常情况下，你无需编辑此文件，除非你要作位置指定的配置文件。
- 个人配置文件(**.acar.cfg**) – 使用 **Settings → Save A/Car Configuration**。

练习 1 打开并运行一个装配


■ 问题阐述

- ◆ 本练习中，你将首次接触到两个 **ADAMS/Car** 中比较典型的仿真。基本上，**ADAMS/Car**进行或者悬架或者进行整车的分析。此处，你将分别进行两种分析：一个整车的 **ISO** 变线分析和一个前悬架的同向双轮跳分析。你还将添加一个轨迹标记点以观察模型中某些部件的运动情况。



本练习大约需要半个小时完成。

练习 1 打开并运行一个装配

- 设置你的进程
- 要生成一个工作目录：
 - ◆ 据你所用的操作系统，分别进行下面的操作：
 - **Windows**系统：
 - 在硬盘上建立一个新的名为 **acar** 目录，比如：**C:\acar**。
 - **UNIX**操作系统：
 - 要在你你的工作目录下启动**ADAMS/Car**，打开一个**UNIX**的命令窗口，并输入 **cd** 命令。
 - 要生成一个名为 **acar** 的目录，输入 **mkdir acar** 的命令。
 - 要进入该新目录，输入 **cd acar** 命令。
- 要启动ADAMS/CAR：
 - ◆ 据你所用的操作系统，分别进行下面的操作：
 - **Windows**系统，从 **Start Programs MSC.Software MSC.ADAMS 2005 ACar ADAMS – Car**。
 - 在 **UNIX** 命令窗口，输入 **adams05**。
 - 从工具条中选择工具按钮 **ADAMS/Car** 
 - 出现 **Welcome** 对话框。

练习 1 打开并运行一个装配

■ 要切换到 **Standard Interface** :

- ◆ 从 **Welcome** 对话框内，选择 **Standard Interface**，然后选择**OK**。
(有时 **Welcome** 对话框内包含选择模式的选项，但有时不包括，这主要取决于配置文件中的设置)。要进行仿真，你必须在**Standard Interface** 模式下。
- ◆ 一旦进入了 **ADAMS/Car** 进程，你可以通过菜单 **Tools** 在不同的模式间切换。如果 **ADAMS/Car** 的**Template Builder** 处于列出状态，表明你现在正处于 **Standard Interface** 模式。

练习 1 打开并运行一个装配

■ 要设置工作目录：

1. 从菜单 **File** 下选择 **Select Directory**。
2. 选择你刚刚建立的 **acar** 目录。
3. 选择**OK**。

ADAMS/Car 设置要存储你输出文件的工作目录。

■ 要生成一个新的数据库并将其设为缺省的可写数据库：

1. 从菜单 **Tools** 下指向 **Database Management**，然后再选择**Create Database**。

生成新数据库的对话框 **Create New Database** 出现。

2. 在 **Database Alias** 文本栏内输入 **acar_training**。
3. 在 **Database Path** 文本栏内输入所要的路径。数据库的名称为数据库路径的别称，需要明确给出。例如：
 - 在**Windows**系统上：**c:\acar\acar_training.cdb**
 - 在**UNIX**系统上：**/acar/acar_training.cdb**

练习 1 打开并运行一个装配

4. 选择 **OK**。

此操作将生成目录并将其添加到你的 **ADAMS/Car** 进程数据库的搜索列表中。

5. 要验证此数据库是否已添加到数据库列表中，可以从菜单 **Tools** 中指向 **Database Management** 然后再选择 **Database Info**。但此时此数据库尚未添加到你的个人配置文件 **.acar.cfg** 中，因此，下一次，在你打开 **ADAMS/Car** 时，它并不会加到搜索列表内。
6. 从菜单 **Tools** 下指向 **Database Management** 然后选择 **Set Default Writable**。
7. 确认 **Database Name** 被设置为 **acar_training** (选择向下箭头再选择 **acar_training**)。
8. 选择 **OK**。

ADAMS/Car 自动地存储到配置文件。

练习 1 打开并运行一个装配

■ 仿真一个整车装配

- ◆ 你首先打开一个整车的装配模型并利用驾驶机器模型进行一下整车的ISO变线分析，然后通过动画观察一下仿真结果。动画过程基于你的分析结果。

■ 要打开一个整车装配：

1. 从菜单 **File** 指向 **Open** 再选择 **Assembly**。
2. 在 **Assembly Name** 文本栏内点击鼠标右键并指向 **Search** 再选择 **<acar_shared>\assemblies.tbl**，然后再选择**MDI_Demo_Vehicle.asy**。
3. 选择 **OK**。

当装配模型打开后，ADAMS/Car 在信息窗口 **Message** 内给你通知提示。显示在 **Message** 窗口中的内容是很重要的，在后面的章节内我们还会详细讨论。

4. 关闭 **Message** 窗口。


练习 1 打开并运行一个装配

■ 要进行仿真：

1. 从菜单 **Simulate** 下指向 **Full-Vehicle Analysis**再指向 **Course Events**，然后选择 **ISO Lane Change**。
2. 在 **Output Prefix** 文本栏内输入 **workshop1a**。
3. 在 **Initial Velocity** 文本栏内输入 **70**。缺省的单位应该为 km/h。
4. 选择 **OK**。
在 Message 窗口内，ADAMS/Car 提示你分析的进程以及仿真完成等。
5. 关闭 **Message** 信息窗口。

练习 1 打开并运行一个装配

■ 要观察仿真结果：



1. 从菜单 **Review** 下选择 **Animation Controls**。
2. 要动画显示车辆装配模型的运动过程，选择按钮 **Play** 。
3. 放大以观看到更多的路面网格：
 - 在键盘上点击 **z**。
 - 按住鼠标左键上下移动，直至显示合适为止。
 - 要退出缩放模式，释放鼠标按钮。

■ 要观察车辆的运动轨迹：

1. 在 Animation Controls 对话框内，将 **No Trace** 改为 **Trace Marker**。
2. 在 **Trace Marker** 一栏内点击鼠标右键 **Marker** 在选择 **Browse**。
Database Navigator 对话框显示出来。
3. 在 **MDI_Demo_Vehicle** 双击鼠标左键再选择 **TR_Body** 再选择 **ges_chassis**，然后选择 **cm**。
4. 选择 **OK**。

练习 1 打开并运行一个装配

■ 要在动画过程中始终跟踪汽车：

1. 改变 **Fixed Base** 为 **Base Part**。
 2. 在 **Base Part** 文本栏内点击鼠标右键指向 **Body**，然后再选择**Pick**。
 3. 在车辆上移动鼠标并点击鼠标右键，可以看到在鼠标旁边显示一系列的部件列表。
 4. 选择车辆上的任意一个部件，比如：选择 **ges_chassis**。
 5. 选择 **Play** 工具 。
 6. 关闭 **Animation**  **ntrols** 对话框。
- 摄像头应该随着车辆移动，同时在车辆的后面出现标记点的白色轨迹线。

练习 1 打开并运行一个装配

- 要从驾驶员位置全景观察整车的动画：
 1. 从菜单 **Review** 下选择 **Postprocessing window** 或直接按 **F8**。
 2. 在视窗 **viewport** (主窗口)内点击鼠标右键并选择 **Load Animation**。
Database Navigator 打开。
 3. 选择 **workshop1a_ilc** , 然后再选择 **OK**。
ADAMS/PostProcessor 加载动画到视窗内。
 4. 按键盘上的 **Ctrl + f** 以在窗口内显示全模型。
 5. 按键盘上的 **c** 键并选择车辆上的一个点作为新的旋转模型视图的中心。
 6. 要旋转车辆模型，以便从车后面观察车辆的运动，按键盘上的 **r** 键，并且按住鼠标左键，移动鼠标直至旋转到合适的位置。

练习 1 打开并运行一个装配

7. 在工具 **Dynamic Translate (xy)** 上点击鼠标右键。
8. 选择按钮 **Dynamic Translate (z)**。
9. 按住鼠标左键并移动鼠标以移动车辆到合适的位置。
10. 选择 **Camera** 按钮。
11. 在 **Follow-Object** 文本栏内点击鼠标右键并指向 **Part** 再选择 **Pick**。
12. 选择车辆上的任意一个部件。
注意：如果你选择了转向轮，略过第13步。
13. 要锁定你在第12步所选择的部件作为观察点。选择 **Lock Rotations**。
14. 要从新的全景显示动画，选择工具按钮 **Play**。
15. 按 **F8** 键返回到 ADAMS/Car。

练习 1 打开并运行一个装配

■ 仿真一个悬架装配

- ◆ 与你仿真整车装配的方式一样，再仿真一个悬架装配。

■ 要打开一个旋架装配：

1. 从菜单 **File** 下指向 **Open**，再选择 **Assembly**。
2. 在 **Assembly Name** 文本栏内点击鼠标右键，再指向 **Search**，选择 **<acar_shared>\assemblies.tbl** 再选择 **mdi_front_vehicle.asy**。
3. 选择 **OK**。
在 **Message** 窗口，**ADAMS/Car** 提示你模型已经准备就绪。
4. 关闭 **Message** 窗口。

练习 1 打开并运行一个装配

■ 要进行一次悬架的同向双轮跳分析：

1. 从菜单 **Simulate** 下指向 **Suspension Analyses** 再选择 **Parallel Wheel Travel**。

2. 按照下面设置分析：

- **Output Prefix:** workshop1b
- **Number of Steps:** 10
- **Bump Travel:** 100
- **Rebound Travel:** -100

车轮上下跳动距离缺省的单位应该是 mm (Settings → Units).

3. 选择 **OK**。

4. 仿真结束后，关闭 **Message** 窗口。

练习 1 打开并运行一个装配

■ 要通过动画观察模型仿真结果：

1. 从菜单 **Review** 下，选择 **Animation Controls**。
2. 改变 **Base Part** 为 **Fixed Base**。
3. 改变 **Trace Marker** 为 **No Trace**。
4. 选择 **Play**。

■ 要观察车辆上一个标记点的轨迹：

1. 改变 **No Trace** 为 **Trace Marker**。
2. 在 **Trace Marker** 文本栏内点击鼠标右键并指向 **Marker**，并选择 **Browse**。
Database Navigator 出现。
3. 选择 **MDI_FRONT_SUSPENSION**，再选择 **gel_spindle**，然后再选择 **cm**。
4. 选择 **Play**。
5. 缩放视图以观察标记点 **gel_spindle.cm** 的白色运动轨迹。

第二章 基本概念

- 在本章，你将学习如何在ADAMS/Car中生成一个模型。重点介绍定义模型的三个基本概念：模版、子系统和装配之间的联系与区别。

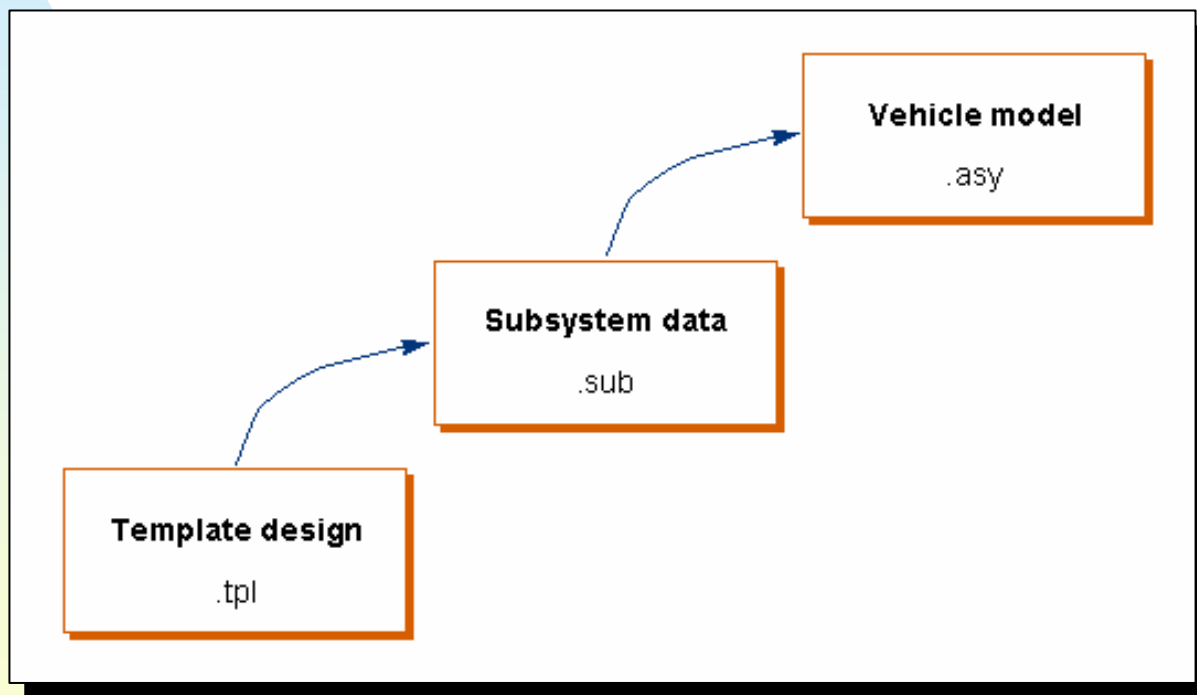
基本概念

- 本章内容：
 - ◆ 数据的谱系关系
 - ◆ 试验台
 - ◆ 主要角色与次要角色
 - ◆ 命名规则

数据的谱系关系

- 建立车辆模型的文件的三个层次 (整车或二分之一车) :
 - ◆ 模版(**Template**) – 定义车辆子装配的拓扑结构 (亦即, 部件和约束是怎样组合在一起成为一个模型的, 需要与其它子系统交换哪些信息, 等等)。例如, 一个模板能够被定义为悬架类模板, 可以用来定义或者前悬或者后悬。
 - ◆ 子系统(**Subsystem**) – 参考模板的一个机械系统模型, 通过提供参数以调整模板进行剪裁 (例如, 定义部件尺寸的位置和弹簧刚度等)。通常来讲, 此模型为你车辆系统中的一个主要子系统, 比如, 前悬架、转向系统或车身等。一个子系统可以看作是对应模板的一个特例, 其中, 用户定义了新的硬点位置和特性文件等。
 - ◆ 装配(**Assembly**) – 一系列的子系统和一个试验台组装在一起构成一个车辆或悬架的模型。模型中, 试验台提供必要的激励, 以便进行各种分析。

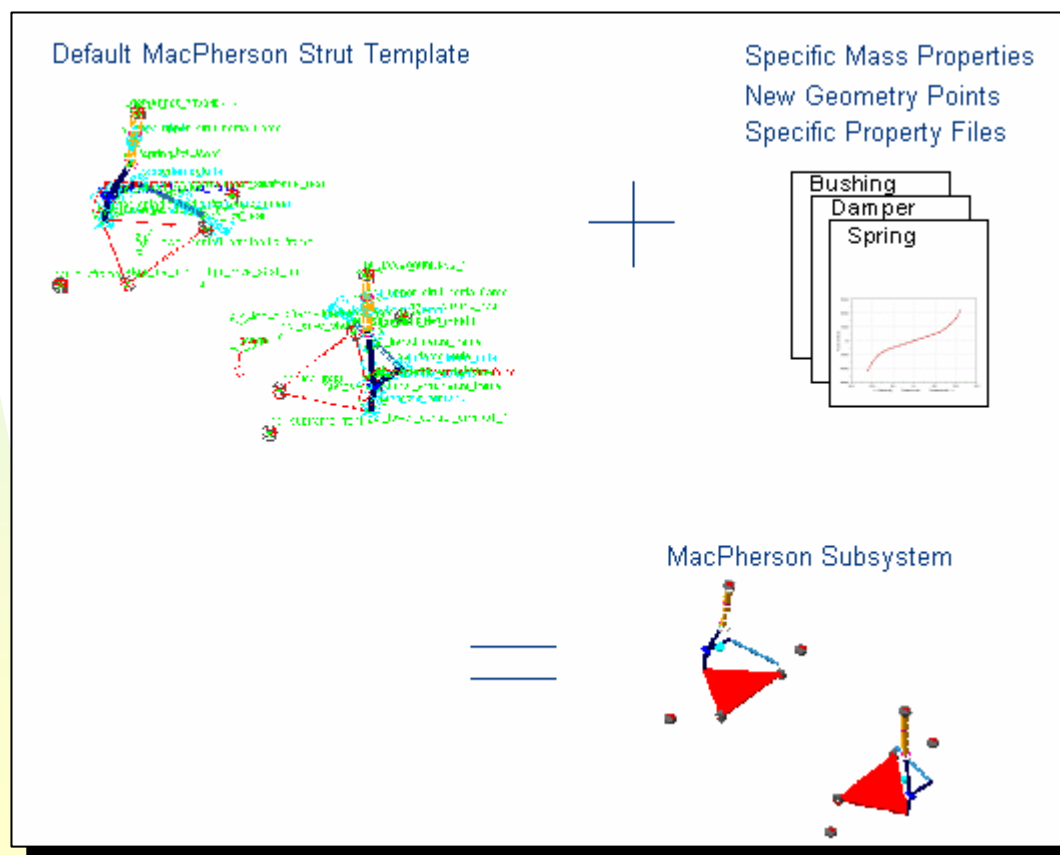
数据的谱系关系



数据的谱系关系

- ◆ 下图表示如何从一个模板生成一个子系统。模板中保留了缺省的几何和拓扑关系，而子系统为模板的一个特例，其中定义新的模型参数，如硬点位置、特性文件和质量性质等。

子系统文件生成



数据的谱系关系

- 通过选择 Tools → Show File (mdi_front_vehicle.asy), 你可以看到一个装配的谱系关系：

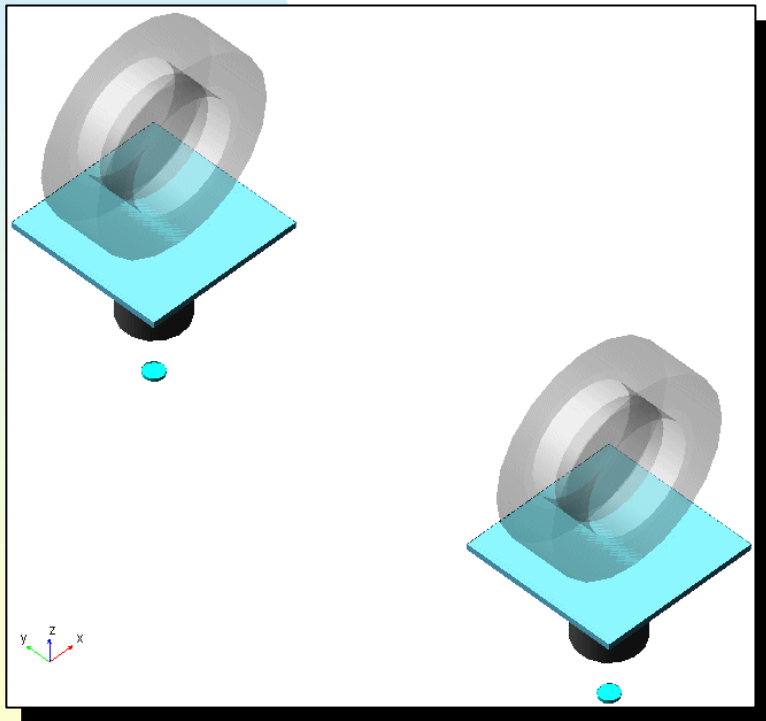
```
$-----MDI_HEADER
[MDI_HEADER]
FILE_TYPE      = 'asy'
FILE_VERSION   = 1.0
FILE_FORMAT    = 'ASCII'
HEADER_SIZE    = 9
(COMMENTS)
{comment_string}
'ADAMS/Car suspension assembly'
$-----ASSEMBLY_HEADER
[ASSEMBLY_HEADER]
ASSEMBLY_CLASS = 'suspension'
TIMESTAMP      = '2002/09/26,14:26:53'
HEADER_SIZE     = 5
$-----UNITS
[UNITS]
LENGTH        = 'mm'
FORCE          = 'newton'
ANGLE          = 'deg'
MASS           = 'kg'
TIME           = 'sec'
$-----SUBSYSTEM
[SUBSYSTEM]
$ Subsystem information:
$   Major Role : suspension
$   Minor Role : front
$   Template   : _macpherson
USAGE = 'mdids://acar_shared/subsystems.tbl/MDI_FRONT_SUSPENSION.sub'
$-----SUBSYSTEM
[SUBSYSTEM]
$ Subsystem information:
$   Major Role : steering
$   Minor Role : front
$   Template   : _rack_pinion_steering
USAGE = 'mdids://acar_shared/subsystems.tbl/MDI_FRONT_STEERING.sub'
...
```


试验台

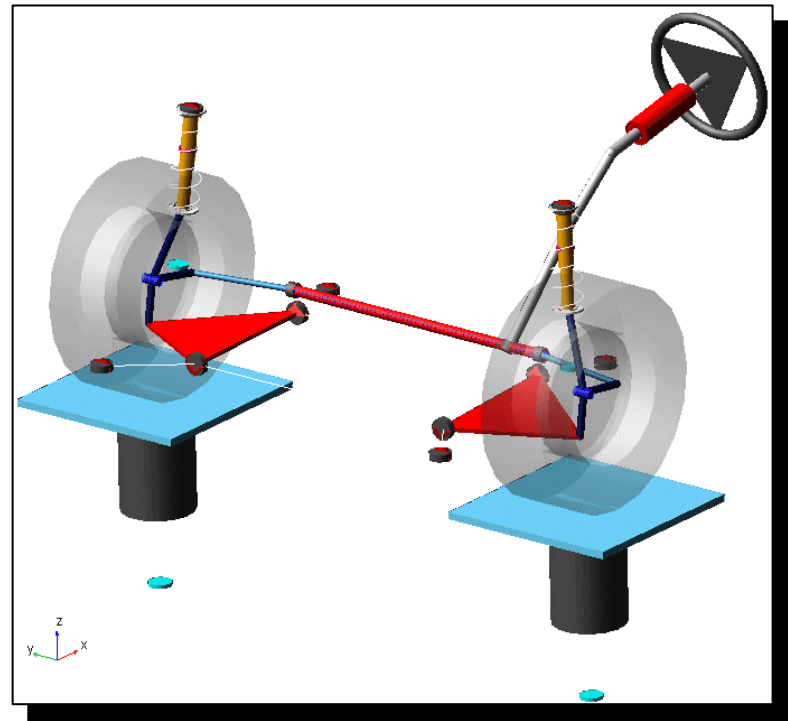
- ADAMS/Car中的试验台为模型的组成部分，主要提供车辆上的各种强制运动。根据不同的模型和仿真工况，需要使用不同的试验台。
- 试验台，可以说是一个特殊的子系统，与其它所有的子系统相连接组成你的模型，构成一个装配分析的模型。左图显示一个单独的悬架试验台，加上悬架子系统，组合起来的系统看起来应该如右图所示。

试验台

单独的悬架试验台



悬架试验台与悬架子系统和转向子系统一起组装成为一个悬架分析装配



主要角色与次要角色

- ◆ ADAMS/Car 使用主要(*major*)和次要(*minor*)角色生成一个正确的装配。主要角色与次要角色用来定义一个子系统在装配中的位置。
- ◆ 任何模板(以及由模板衍生而来的子系统)都需要有一个主要角色的定义, 如: suspension、steering、body、anti-roll bar、wheel等等。
- ◆ 当生成一个子系统时, 标准用户需要赋予子系统一个次要角色, 如: *front*、*rear*、*trailer*或*any*。这样做的目的就是可以使用同一模板分别定于或前或后的悬架子系统。
- ◆ 要生成一个正确的悬架装配, 最少需要一个悬架子系统和ADAMS/Car的悬架试验台(suspension test rig)。
- ◆ 要生成一个正确的车辆装配, 最少需要一个前悬架子系统、一个后悬架子系统、前、后车轮子系统、一个车身子系统和一个转向子系统加上一个ADAMS/CAR车辆试验台。

主要角色与次要角色

- 要增加更多的主要角色与次要角色，可以使用一个 在你工作目录下的 acarBS.cmd 文件，文件中可以包含下面的命令：

```
Major roles
```

```
!-- Add a new major role ----
```

```
variable modify variable=.ACAR.variables.major_roles &  
string_value=(eval(.ACAR.variables.major_roles)), "subframe"
```

```
Minor roles
```

```
!-- Add a new minor role ----
```

```
variable modify variable=.ACAR.variables.minor_roles &  
string_value=(eval(.ACAR.variables.minor_roles)), "rear2"
```

命名规则

- ADAMS/Car中所有的对象都按照一个命名规则进行命名。一个给定的对象的前三个字母体现该对象的类型和几何对称性。
- 例子：
 - ◆ gel_arm: General_Part_Left_....
 - ◆ hps_lcs_front: Hard_Point_Single_...
 - ◆ bkl_mount: Bushing_Kinematic_Left_...
 - ◆ nsr_main_spring: Non-linear_Spring_Right_...
 - ◆ pvs_toe_angle: ParameterVariable_Visible_Single_...



完整列表，参见 ADAMS/Car 的在线帮助：
Components tab → About the Naming Convention.

练习 2：模板 VS. 子系统

■ 问题阐述

- ◆ 理解 ADAMS/Car 中模板和子系统的区别是用好ADAMS/CAR的第一步。举例来说，两个人一起工作，都搞转向系统。看看他们的计算机屏幕，似乎是同一个模型，但是，区别很大，一个是使用模板模式建立模板，而另一个是使用标准模式建立子系统，有什么区别？
- ◆ 正如前面所述，区别在于：你在不同的模式下能做什么？拓扑关系或部件之间的连接方式等信息是在模板建模模式下通过参数(变量)来定义的；而具体定义这些参数则需要到标准建模模式下进行。另外，你只能在标准建模模式下进行仿真，基于在模板建模模式下所生成的模型。要更好的或更准确的理解二者之间的区别所在，可以分别生成一个模板和一个子系统文件，再比较文件中内容的不同之处。



本练习大约需要半个小时完成。

练习 2：模板 VS.子系统

■ 打开一个模板文件

- ◆ 在本练习中，你将生成一个 ASCII 格式的模板文件并与一个子系统文件进行比较。模板文件既可以保存为二进制格式文件，也可以保存为 ASCII 格式的文本文件。缺省情况下，为二进制格式文件，因此要查看文件中的内容，你必须将其保存为文本文件。

■ 要在ADAMS/CAR中选择模板建模模式：

- ◆ 从菜单 **Tools** 下选择 **ADAMS/Car Template Builder**。

注意：通过简单的按功能键 F9你就可以在不同的建模模式间转换。

练习 2：模板 VS.子系统

- 如果在 ADAMS/Car 的菜单 Tools 下没有出现模板建模模式的选项，你必须设置用户配置文件：
 1. 退出 ADAMS/Car。
 2. 修改你的用户配置文件 .acar.cfg，其中包含：
ENVIRONMENT MDI_ACAR_USERMODE expert
*提示：*有关 .acar.cfg 文件的存放位置，请参见第一章 存储配置文件部分。
 3. 再次启动 ADAMS/Car。
 4. 参考练习一的相关内容，设置工作目录为 acar。

练习 2：模板 VS. 子系统

- 要打开共享数据库下的 MacPherson 式悬架模板：
 1. 从菜单 **File** 下选择 **Open**。
 2. 在 **Template Name** 文本栏内点击鼠标右键，指向 **Search**，然后选择：**<acar_shared>/templates.tbl**。
 3. 双击 **_macpherson.tpl**。
 4. 选择 **OK**。

练习 2：模板 VS. 子系统

■ 要存储模型为文本格式：

1. 从菜单 **File** 下选择 **Save As**。
2. 在 **New Template Name** 文本栏内输入 **mac_ascii**。(注意此文本栏为灰色的，因为执行此操作中此项并非必须输入的项目。如果你不输入名字的话，ADAMS/Car 按照当前模型的名字存储。)
3. 设置 **File Format** 为 **Ascii**。此操作是必须的，只有这样，所生成的文件才可读。
4. 选择 **OK**。

ADAMS/Car 存储文件在数据库 **acar_training** 下，此数据库是你在上一个练习中所设定的。

练习 2：模板 VS.子系统

■ 要打开此文件并察看一下文件中的内容：

1. 使用文本编辑器打开文件 **_mac_ascii**。

对于你所在的操作系统不同，_mac_ascii文件所处的位置分别为：

- Windows系统: C:/acar/acar_training.cdb/templates.tbl
 - UNIX系统: /acar/acar_training.cdb/templates.tbl
2. 如果你没有将 acar_training 设置为缺省的可写数据库的话，请使用 **Tools → Database Management → Database Info** 找到缺省的可写数据库的别名。

练习 2：模板 VS. 子系统

- ◆ 此文件中包含定义模型所有的信息，如标记点、部件、信息交换器、载荷等等。看一下文件中所存储的信息。例如，下面为定义模型中开头的部分(在顶部)：

```
!  
!----- gel_lower_control_arm -----!  
!  
!  
defaults coordinate_system &  
  default_coordinate_system = ._mac_ascii.ground  
!  
part create rigid_body name_and_position &  
  part_name = ._mac_ascii.gel_lower_control_arm &  
  location = 0.0, -550.0, 150.0 &  
  orientation = 0.0d, 90.0d, 180.0d
```

- ◆ 注意生成部件、约束、标记点等的命令。你是否能够完全理解文件中所有的命令并不重要，重要的是，你先有一点感觉，知道此文件中所存储的信息类型就可以了。

练习 2：模板 VS.子系统

■ 打开一个子系统文件

- ◆ 子系统文件只以 ASCII 格式的文本文件存储，因此你无需将二进制的文件转化为文本格式的文件。你不必生成 MacPherson 的悬架系统，你只需打开一个共享数据库下基于 MacPherson 模板所生成的子系统文件即可。

■ 要打开一个子系统文件并察看其内容：

1. 从菜单 **Tools** 下选择 **Show File**。
2. 在 **File Name** 文本栏内点击鼠标右键并指向 **Search** 然后再选择 **<acar_shared>**。
3. 双击目录 **subsystems.tbl** 然后双击 **MDI_FRONT_SUSPENSION.sub**。
4. 选择 **OK**。

在 Information 窗口内显示 MDI_FRONT_SUSPENSION.sub 的内容。

练习 2：模板 VS. 子系统

5. 看一看文件中所存储的信息。

文件的最顶部看起来与模板文件很相似，但下面的就完全不同了。文件中重置了模板文件中的参数。

注意在[SUBSYSTEM_HEADER] 部分：

TEMPLATE_NAME = '<acar_shared>/ templates.tbl/_macpherson.tpl'。

同样要注意子系统中参数的设置

```
$-----PART_ASSEMBLY
[PART_ASSEMBLY]
USAGE      = 'lower_control_arm'
SYMMETRY   = 'left/right'
MASS       = 5.0911573156
$ Part location is dependent.
$ X,Y,Z location= -6.6666666667, -496.6666666667, 225.0
$ Part orientation is dependent.
$ ZP vector   = -0.0652566755, -0.9951643011, -0.0734137599
$ XP vector   = -0.9972332421, 0.0676631757, -0.030782389
CM_LOCATION_FROM_PART_X = 0.0
CM_LOCATION_FROM_PART_Y = 0.0
CM_LOCATION_FROM_PART_Z = 0.0
IXX          = 26908.978153
IYY          = 60577.701004
IZZ          = 33765.551131
IXY          = 0.0
IZX          = 3142.3266008533
IYZ          = 0.0
```

练习 2：模板 VS.子系统

■ 总结

- ◆ 总的说来，模板定义一个模型的拓扑结构，而一个子系统再确定参数以生成一个模板的特例。
- ◆ 下表列出了两类文件的特性：

表 1：模板和子系统的比较

特征：	模板：	子系统：
用于定义模型的框架	是	否
参照其它文件 (模板或子系统)	否	是
可以编辑以改变拓扑关系(例如，载荷的作用点)	是	否
修改定义模型的参数	是	是
用于直接定义一个装配	否	是(在装配中)

练习 2：模板 VS. 子系统

- ◆ ADAMS/Car的模板是由具有 ADAMS/Car 丰富使用经验的用户使用 ADAMS/CAR模板模式建立的 MSC.ADAMS 模型。ADAMS/Car 的模板模式包括几何和拓扑数据，模板文件可以以 ASCII 文本或二进制格式存储。
- ◆ ADAMS/Car 的子系统是基于 ADAMS/Car 的模板建立的，允许标准用户改变几何数据和模板中的部分数据，子系统文件。
- ◆ ADAMS/Car的装配是由一系列的 ADAMS/Car 的子系统加上一个 ADAMS/Car 的试验台组装而成，装配文件是以 ASCII 文本格式存储的，包括一系列子系统的列表和与装配有关的试验台。

第三章 生成并调整子系统

- 在本章，你将学习如何从一个模板生成子系统，同时还要了解在子系统中哪些参数你可以调整。

生成并调整子系统

■ 本章内容：

- ◆ 生成子系统
- ◆ 修改子系统
- ◆ 调整硬点
- ◆ 调整参数变量
- ◆ 调整质量特性
- ◆ 将刚性体替换为弹性体
- ◆ 调整弹簧和阻尼器
- ◆ 对子系统其它可用的调整
- ◆ 实体(Instance)定义的替换

生成子系统

- 要生成一个新的子系统，必须先有模板
- 因为你只能在标准界面下生成子系统，所以确认你是在标准界面下。从菜单 File 下指向 New 然后选择 Subsystem。
- 在 New Subsystem 对话框内，填写下列文本栏：
 - ◆ Subsystem Name
 - ◆ Minor Role
 - ◆ Template Name
 - ◆ Translation values (可选参数，但不能侧向移动模板的位置)

修改子系统

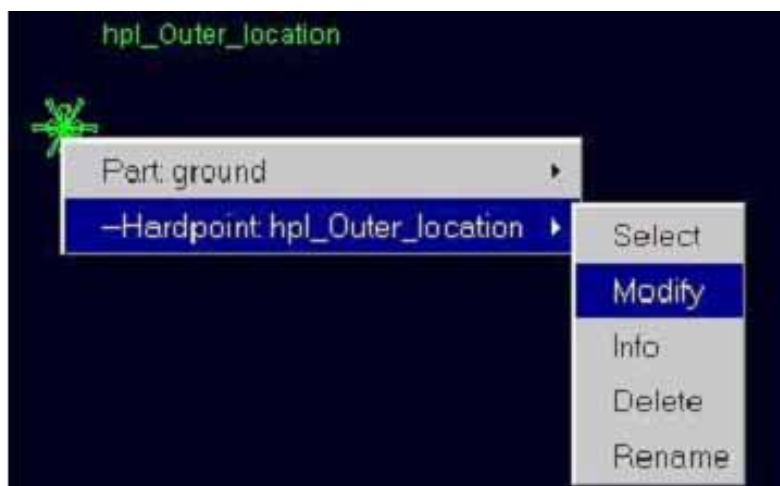
- 在本章，我们将涉及到下面一些可以对子系统的修改：
 - ◆ 调整硬点
 - ◆ 调整参数变量
 - ◆ 调整质量特性
 - ◆ 调整弹簧和阻尼器
 - ◆ 对子系统其它可用的调整

调整硬点

- 在子系统中，你能够移动硬点的位置，这些硬点的位置在模板已经给出了缺省的数值，你可以对其修改，变为你自己模型下的位置。硬点通常用来定义模型中所有的关键点。有关硬点的更多信息，参见第九章。
- 在标准界面，从菜单 Adjust 选择 Hardpoint，有三个选项你可以选：
 - ◆ **Modify** – 显示一个对话框以选择一个硬点再修改其位置。
 - ◆ **Table** – 显示一个表，表中显示子系统中所有的影点，你可以修改表中任意一个硬点的位置。
 - ◆ **Info** -显示一个对话框以选择实体类型和子系统。缺省的，是已经经过预选为hardpoint类型和当前子系统，结果将给出子系统下所有硬点的信息。

调整硬点

- 在标准界面，在视窗内，在硬点对象上点击鼠标右键，从中选择各种操作。



调整参数变量

- 在子系统中，你可以改变模板中所定义的参数变量的值。一个参数变量，简单地讲，就是一个原来存储关键参数信息的变量。例如，在一个悬架模板中，前束角(toe)和外倾角(camber)或车轴的轴线方向通常用参数变量。注意，参数变量也可以用来存储文字。

调整参数变量

- ADAMS/Car 自动地定义一些参数变量，因为在汽车性能分析中普遍用到这些参数(比如，前束角 toe 和外倾角 camber)。但你可以生成一些新的参数变量。
- 在模板建模模式下，你可以生成一些参数变量，这些参数变量对于标准用户是隐藏的，意味着，标准用户不能修改这些参数变量。隐藏参数变量的命名规则是：

ph[lrs]_(name): ParameterVariable_**H**idden_**L**eft, **R**ight, **S**ingle]

- 当你不希望标准用户修改某些参数变量的值时，你可以使用隐藏参数变量。

调整参数变量

- 要调整参数变量，从菜单 Adjust 下选择 Parameter Variable，你可以有两种选择：
 - ◆ **Modify** – 显示对话框以选择一个参数变量再修改其数值。
 - ◆ **Table** – 显示一个表，表中显示子系统中所有的参数变量，你可以修改表中任何一个参数变量。

调整质量特性

- 在生成模板时，每个部件都被赋予一个缺省的质量特性，你可以在标准界面下修改这些数值。
- 要调整质量特性，从菜单 Adjust 下选择 General Part，你可以有两种选择：
 - ◆ **Modify** – 显示一个对话框，你可以选择一个要修改的部件，再指定其质量、转动惯量等。你也可以通过在部件上点击鼠标右键再选择所出现的 Modify。
 - ◆ **Calculate Mass** - ADAMS/Car 可以基于一个部件的几何外形和密度来计算其质量和转动惯量特性。注意，如果在标准界面下部件的几何外形与缺省的发生了变化，相应的质量特性并不自动刷新。要刷新的话，简单的再次计算一下质量特性即可。如果你的几何外形是从CAD软件中导入的，并且非常复杂，你可能需要手工输入质量特性。

将刚性体替换为弹性体

- 你可以将刚性体替换为弹性体或将弹性体替换为新的弹性体，并且可以帮助你选择正确的外接连接点的节点号。
- 在此版本之前，如果你想在模型中包含弹性体的话，你必须要重新生成新的模板。现在，模拟弹性成了MSC.ADAMS 部件的属性，因此，引入弹性体成为标准界面下的操作，并且反映到子系统的文件中。

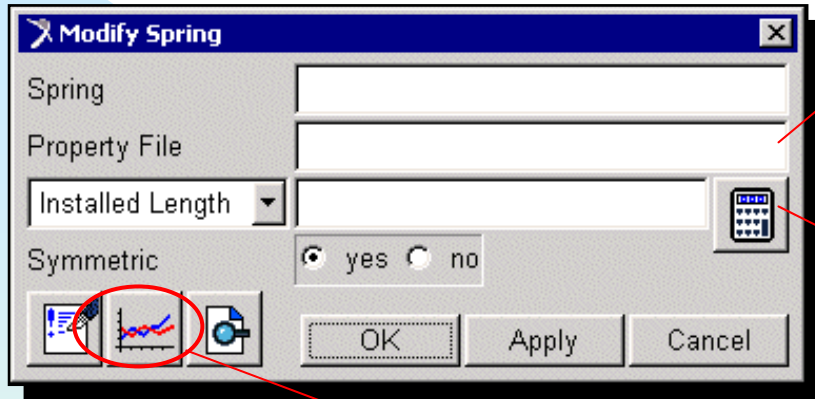
将刚性体替换为弹性体

- 在标准界面下直接引入弹性体有如下的诸多好处：
 - ◆ 避免多个模板
 - ◆ 避免使用多余的哑物体(没有质量的中间部件)来保证与模型中其它部件的正确连接。
 - ◆ 原来已有约束的自动替换
 - ◆ 一些高级的工具，可以帮助从刚性体到弹性体替换时连接点的探测和替换。
- 要使用这一功能，使用在部件修改对话框内的 Rigid to Flex 按钮。

调整弹簧和阻尼器

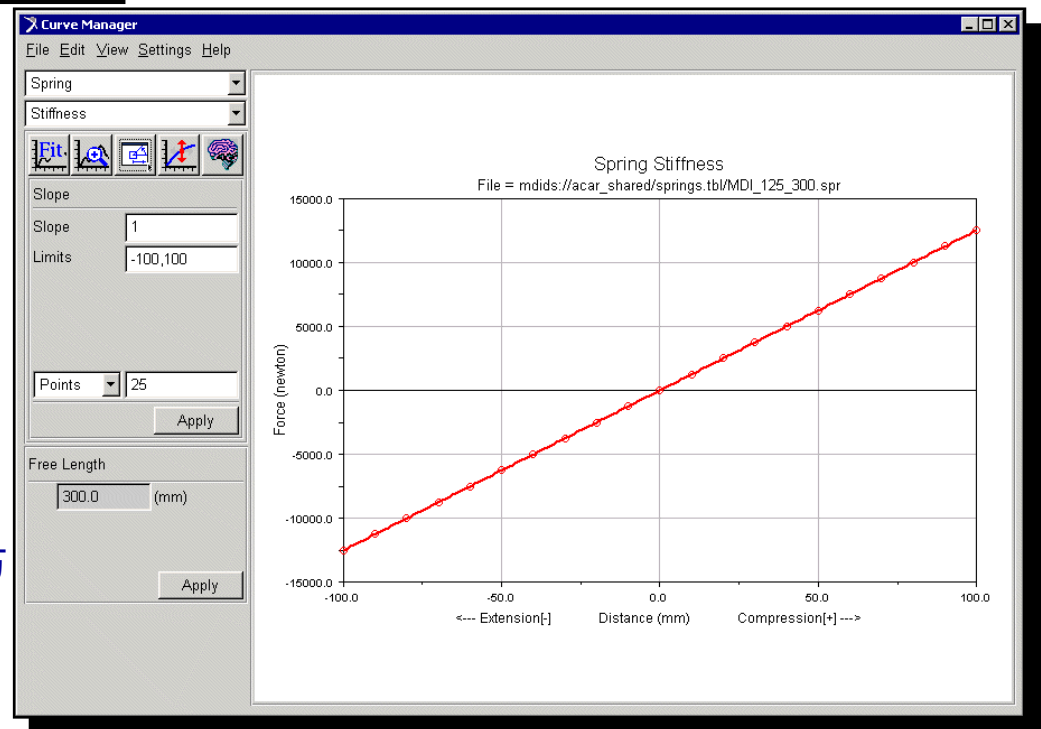
- 弹簧和阻尼器是在模板建模模式下定义的，并且参考位于你所选择的数据库下的特性文件。在标准用户界面下，你可以将弹簧和阻尼器的特性文件重现定位或直接重新生成新的特新文件。
- 要调整弹簧和阻尼器，在要调整弹簧和阻尼器上点击鼠标右键并选择 Modify，将出现下面的对话框：

调整弹簧和阻尼器



指定描述该弹簧特性文件的路径

计算满足给定预载荷所需要的安装长度。



关于 Curve Manager，参见后面的章节

调整弹簧和阻尼器

- 在 Modify Spring 的对话框内，当你在 property file 文本栏内点击鼠标右键时，ADAMS/Car 会自动的转到你所选数据库下的 spring.tbl 目录 (类似地，当你在 Modify Damper 对话框内的property file 文本栏内点击鼠标右键时，ADAMS/Car 会自动的转到你所选数据库下的 damper.tbl 目录)。

对子系统其它可用的调整

- 更多选项，选择 View → Subsystem
- 悬架子系统(Suspensions)
 - ◆ 是否考虑传动系(Toggle driveline activity on/off)
 - ◆ 悬架参数数组(Suspension parameter array) (参见第10章)
 - ◆ 前束角/外倾角(Toe/camber values)
- 转向子系统(Steering)
 - ◆ 传动比(Gear ratios)
 - ◆ 转向辅助特性文件(Steering assist property file)
- 传动子系统(Powertrain)
 - ◆ 传动系特性文件(Powertrain map property file)
 - ◆ 差速器特性文件(Differential property file)

实体(Instance)定义的替换

- 修改元件的另一种途径使用同类型的元件替换现有元件的定义。例如，你可以将一个圆弹簧替换为一个空气弹簧。下面介绍如何修改一个元件的定义。按照类似的操作步骤，你可以对其它类型的元件进行替换。
- 现在软件支持的元件包括：空气弹簧(air spring)、衬套(bushing)、阻尼器(damper)、弹簧(spring)以及一些专用的元件。

注意：你只能在标准界面下改变元件的定义。

实体(Instance)定义的替换

■ 要改变元件的定义：

1. 在标准界面下，在某个元件上点击鼠标右键，指向其名字，然后选择 **Replace Instance**。
2. 按 **F1** 键然后按照对话框中的指导完成实体(Instance)定义的替换
3. 选择 **OK**。

练习 3 生成并调整子系统

■ 问题阐述

- ◆ 在本练习中，你将生成一个新的子系统并学习如何调整其参数。



本练习大约需要半个小时完成。

练习 3 生成并调整子系统

■ 生成并存储子系统

■ 要生成子系统：

1. 转到 ADAMS/Car 标准界面下。
 2. 从菜单 **File** 下指向 **New**，然后选择 **Subsystem**。
 3. 在 **Subsystem Name** 文本栏内输入 **my_macph**。
 4. 设置 **Minor Role** 为 **rear**。
 5. 在 **Template Name** 文本栏内点击鼠标右键并指向 **Search**，再选择 **<acar_shared>\templates.tbl**。
 6. 双击 **_macpherson.tpl**。
 7. 选择 **OK**。
- ADAMS/Car 显示悬架子系统。

练习 3 生成并调整子系统

■ 要存储子系统：

1. 从菜单 **File** 下指向 **Save As** 再选择 **Subsystem**。

显示存储子系统的对话框 Save Subsystem。由于在 ADAMS/Car 中只有打开了一个子系统，因此，缺省的 my_macph 被选了，但是，如果你同时打开了若干个子系统，可以使用上下箭头选择你想要存储的子系统。ADAMS/Car 存储子系统在数据库 acar_training 下，即缺省的数据库，对应文件的目录为 subsystems.tbl 内。

2. 选择 **OK**。

练习 3 生成并调整子系统

- 调整子系统
- 要打开你的子系统文件：
 1. 从菜单 **Tools** 下选择 **Show File**。
 2. 在 **File Name** 文本栏内点击鼠标右键，再指向 **Search**, 然后选择 **<acar_training>**。
 3. 双击 **subsystems.tbl**, 再双击 **my_macph.sub**。
 4. 选择 **OK**。

练习 3 生成并调整子系统

■ 快速的浏览一下文件的内容：

1. 浏览一下文件并注意所存储的信息类型。你能看到硬点、弹簧刚度等的参数，这些参数是基于模板里面所定义的数值。要剪裁其参数，使之成为你的特定的一个 MacPherson 式悬架，你需要在标准用户界面下改变这些参数值。
2. 例如，注意下拉臂(LCA)几何外形的厚度，为 10.6823911464 (离顶部大约 1/4 处)。你将在标准界面下改变此值，并在你的子系统文件中更新该值。

```
$-----ARM_GEOMETRY
[ARM_GEOMETRY]
USAGE      = 'lower_control_arm'
PART       = 'lower_control_arm'
SYMMETRY   = 'left/right'
THICKNESS  = 10.6823911464
```

练习 3 生成并调整子系统

3. 检查一下下拉臂(LCA)的质量特性：

```
$-----PART_ASSEMBLY
[PART_ASSEMBLY]
  USAGE                      = 'lower_control_arm'
  SYMMETRY                   = 'left/right'
  MASS                       = 5.0
$ Part location is dependent.
$  X,Y,Z location            = 0.0, -550.0, 150.0
$ Part orientation is dependent.
$  ZP vector                 = 0.0, -1.0, 0.0
$  XP vector                 = -1.0, 0.0, 0.0
  CM_LOCATION_FROM_PART_X    = 0.0
  CM_LOCATION_FROM_PART_Y    = 0.0
  CM_LOCATION_FROM_PART_Z    = 0.0
  IXX                        = 33380.880617
  IYY                        = 25047.547284
  IZZ                        = 58333.333333
  IXY                        = 0.0
  IZX                        = 0.0
  IYZ                        = 0.0
```

4. 选择 **Clear**, 再关闭 **Information** 窗口。

练习 3 生成并调整子系统

■ 要改变下拉臂(LCA) 的厚度：

1. 在任意一个(左/右)下拉臂(LCA)上点击鼠标右键。(红色三角区域)
2. 指向 **Arm: graarm_lower_control_arm**, 再选择 **Modify**。
3. 将 **Thickness** 改为 **33.3**。
4. 选择 **OK**。

在拟定屏幕上，注意 ADAMS/Car 更新了下拉臂(LCA) 的厚度。

■ 要存储你的子系统：

1. 从菜单 **File** 下指向 **Save**。
2. 选择 **Yes** 以生成备份文件。

练习 3 生成并调整子系统

■ 打开并浏览改变后的文件：


1. 从菜单 **Tools** 下选择 **Show File**。你的子系统应该再次输入，选择 **OK**。另外也可以通过右键点击再搜索 **<private>\subsystems.tbl\my_macph.sub**。
在顶部，你可以看到拉臂的几何参数已经更新了：

```
$-----ARM_GEOMETRY
[ARM_GEOMETRY]
USAGE      = 'lower_control_arm'
PART       = 'lower_control_arm'
SYMMETRY   = 'left/right'
THICKNESS  = 33.3
```

2. 也要注意下拉臂的质量特性并未改变，即使下拉臂的几何外形改变了。
3. 关闭 Information 窗口。

练习 3 生成并调整子系统

■ 要调整质量特性：

1. 在下拉臂上点击鼠标右键并选择 **Modify**。
2. 要基于新的几何外形计算下拉臂的质量特性，选择 **Calculate Mass** 工具按钮 。
3. 接受缺省的钢作为部件的材料并用来计算质量特性，选择 **OK**。
4. 要接受对部件的改变，选择 **OK**。
5. 从菜单 **File** 下选择 **Save** 再次存储子系统。
6. 要存储备份文件，选择 **Yes**。

练习 3 生成并调整子系统

7. 从菜单 **Tools** 下选择 **Show File**。你的子系统应该再次输入，选择 **OK**。另外也可以通过右键点击再搜索 **<private>\subsystems.tbl\my_macph.sub**。
在顶部，你可以看到拉臂的几何参数已经更新了：

```
$-----PART_ASSEMBLY
[PART_ASSEMBLY]
USAGE           = 'lower_control_arm'
SYMMETRY        = 'left/right'
MASS            = 20.6616944084
$ Part location is dependent.
$ X,Y,Z location = 0.0, -550.0, 150.0
$ Part orientation is dependent.
$ ZP vector      = 0.0, -1.0, 0.0
$ XP vector      = -1.0, 0.0, 0.0
CM_LOCATION_FROM_PART_X = 0.0
CM_LOCATION_FROM_PART_Y = 0.0
CM_LOCATION_FROM_PART_Z = -0.8930636959
IXX              = 137685.25043
IYY              = 315591.19691
IZZ              = 181724.53753
IXY              = 0.0
IZX              = 0.0
IYZ              = 0.0
```

练习 3 生成并调整子系统

■ 要调整硬点：

1. 从菜单 **Adjust** 下指向 **Hardpoint**。注意你可以选择下面两种操作中的一个：
 - ◆ **Modify** – 你一次只能调整一个硬点。
 - ◆ **Table** – 你可以调整所有的硬点。可选的，你可以打开图标的开关(按一下键盘上的小写字母 'v')。右键点击你想要修改的硬点，然后在要修改硬点的名字上选择 **Modify**。
2. 选择 **Table**。
3. 设置 **Display** 为 **Both**。
注意：也许有必要扩大一下对话框以便可以看到全部的硬点。

练习 3 生成并调整子系统

■ 要编辑下拉臂的几何尺寸：

1. 将硬点 **hpl_lca_front** 或 **hpr_lca_front** 的 **loc_x** 从 **-200** 调整为 **-120**。因为这些硬点是几何对称的，改变一个，意味着另一个也自动改变。
2. 选择 **Apply**。
3. 将硬点 **hpl_lca_rear** 或 **hpr_lca_rear** 的 **loc_x** 从 **200**调整为 **120**。因为这些硬点是几何对称的，改变一个，意味着另一个也自动改变。要在表中显示新的值，按 **Enter**。

注意：现在模型中的硬点的位置还没有改变，但你选择了Apply 或 OK 后，这些调整才起作用。

4. 选择 **Apply**。
现在下拉臂变得比原来小了。

练习 3 生成并调整子系统

■ 要关闭硬点列表：

- ◆ 要关闭该表，选择 **OK** 或 **Cancel**。只有当你存储了你的子系统后，ADAMS/Car 才在 .sub 文件中以新的硬点坐标值更新硬点的位置。

*注意：*由于你还没有存储子系统，ADAMS/Car 也就没有更新子系统文件(my_macph.sub)。

■ 要调整参数变量：

1. 从菜单 **Adjust** 下指向 **Parameter Variable**，然后再选择 **Modify**。
2. 要查看一下模板内总共有多少参数变量，可以在 **Parameter Variable** 文本栏内点击鼠标右键，再指向 **Variable**，然后选择 **Guesses**。

现在你能够看到有一个参数变量可供调整，即 pvl_toe_angle (前束角 toe angle)。前束角为整车的纵向中心线与车轮平面和路面交线之间的夹角。ADAMS/Car 表示前束角的单位是度，并且车轮向车身内转的话为正。

练习 3 生成并调整子系统

3. 选择 **pvl_toe_angle**。注意，现在前束角的值为 0.0。
4. 要观察一下前束角变化的影响，在 **Real** 文本栏内输入 **2.0**。
ADAMS/Car 左右两侧同时更新，因为几何对称性为 yes。
5. 选择 **OK**。

模型中的一个构造坐标系 cf[l,r]_wheel_center (一个标记点)与此参数相关联并会自动地更新，在后面的章节中会详细讨论。



相关信息，参见 ADAMS/Car 在线帮助的 Templates and Analyze 部分。

注意：你能够调整前束角和外倾角，通过选择：

Adjust → Suspension → Parameters → Toe/Camber Values。

练习 3 生成并调整子系统

■ 要调整弹簧：

1. 在弹簧上点击鼠标右键，然后选择 **Modify**。

此处，你可以调整定义弹簧力—变形量的特性文件，同时可以设定安装长度或预载荷的大小，因为二者之间是直接关联的。显示在 Property File 文本栏内的内容是缺省的特性文件
<acar_shared>/springs.tbl/mdi_0001.spr。

2. 要查看弹簧力—变形量(force-deflection)曲线，选择 **Curve Manager** 工具按钮 。

Curve Manager 窗口替代原来的标准界面。注意此弹簧为分段的线性弹簧，因而有两个斜率。

练习 3 生成并调整子系统

3. 要返回到 ADAMS/Car 标准界面，从菜单 **File** 下选择 **Close**。
4. 要改变特性文件，在 **Property File** 文本栏内点击鼠标右键并选择 **Search**，然后再选择 **<acar_shared>\springs.tb\MDI_125_300.spr**。
5. 要查看弹簧力—变形量(force-deflection)曲线，再次选择 **Curve Manager** 工具按钮 。注意此弹簧为线性弹簧。
6. 返回到 ADAMS/Car 标准界面 (File → Close)。
7. 将安装长度设为 **200**。
8. 选择 **OK**。
关于弹簧的定义，参见第 10 章的弹簧部分。


练习 3 生成并调整子系统

■ 要替换阻尼器：

- ◆ 利用 Replace 功能修改阻尼器的种类。

1. 在阻尼器上点击鼠标右键然后选择 **Modify**。

此处你能够调整定义阻尼力—速度曲线的特性文件。显示在 Property File 文本栏内的内容是缺省的特性文件 <acar_shared>/dampers.tbl/mdi_0001.dpr。

2. 要查看阻尼力—速度曲线，选择 **Curve Manager** 工具按钮 。
3. 注意到该曲线是非线性曲线。
3. 返回到 ADAMS/Car 标准界面 (File → Close)。
4. 选择 **OK**。

练习 3 生成并调整子系统

5. 在阻尼器上点击鼠标右键然后选择 **Replace**。
6. 设置 **Definition** 为 **Linear_damper**。
7. 设置 **Replace** 为 **All instances of the same name**。
8. 选择 **OK**。

出现 Modify Linear Damper 对话框。



9. 选择 **View File** 工具。

注意在阻尼特性文件 mdi_0001.lbf 中阻尼比被设为一个常数。

提醒

- ◆ 你此前在标准界面下所作的所有修改都只是影响你的子系统，而不会改变模板。进行这些调整可以有多种方式，而且还要注意的是：你只有在 ADAMS/Car 中存储了之后才改变你的数据库下的子系统文件。

第 4 章 曲线编辑器和特性文件编辑器

- 你可以使用曲线编辑器和特性文件编辑器生成再编辑特性文件中数据，本章正是介绍这部分内容。

曲线编辑器和特性文件编辑器

■ 本章内容：

- ◆ 利用曲线编辑器和特性文件编辑器编制特性文件
- ◆ 特性文件种类
- ◆ 生成特性文件
- ◆ 特性文件编辑器
- ◆ 转换 TeimOrbit 格式文件为 XML格式文件
- ◆ 修改现有的特性文件
- ◆ 绘图 vs. 数表


利用曲线编辑器和特性文件编辑器 编制特性文件

- 特性文件是 ASCII 格式的文件，包含描述元件的数据，比如，阀门弹簧、凸轮或衬套等。因为特性文件都是文本文件，你能够使用任何文本编辑器修改。
- 使用特性文件，你可以：
 - ◆ 在一个模板或子系统内具有同样的特性或参数的多个元件更容易描述。例如，在 ADAMS/Car 中，一个悬架中可能包含很多同样的衬套，这种情况下，所有的衬套的特性可以用同一个特性文件描述。
 - ◆ 不同的模板或子系统也可以共享同样的元件。
- 你可以在不同的模板或子系统内参考相同的特性文件。

利用曲线编辑器和特性文件编辑器 编制特性文件

- 所有的特性文件类型在配置文件 (acar.cfg) 中指定。你修改完特性文件，你可以以现在的名字存储，也可以以新的名字存储。
- 特性文件按照类型分组并存储在相应的数据库子目录下。亦即每种类型的特性文件，比如衬套和阻尼器，就被存储在数据库子目录 bushings.tbl 和 dampers.tbl 下。

利用曲线编辑器和特性文件编辑器 编制特性文件

- 弹簧、阻尼器、止档、反向止档和衬套等均使用一个定义力—位移或力—速度的特性文件子集来定义。对这些元件，你可以使用曲线编辑器和特性文件编辑器生成、编辑和察看其特性文件。你可以从菜单 Tools 下打开曲线编辑器，从此对话框下，你可以使用曲线编辑器或特性文件编辑器编辑特性文件或利用工具按钮  察看特性文件。

特性文件种类

- 曲线编辑器支持下面的曲线类型：
 - ◆ 衬套(Bushing)
 - ◆ 止档(Bumpstop)
 - ◆ 反向止档(Reboundstop)
 - ◆ 弹簧(Spring)
 - ◆ 阻尼器(Damper)
 - ◆ 车轮包络空间(Wheel envelope)
- 曲线编辑器的功能随着所编辑的曲线类型不同而改变。

特性文件种类

■ 曲线类型有两种模式：

- ◆ **绘图(Plotting)** – 在此模式下，你可以指定曲线的函数来生成曲线。
例如，你可以定义一个弹簧的特性曲线，其斜率为 20 N/mm ，变形范围在 -100 到 100 mm 之间，有25个数据点。
- ◆ **数表(Table)** -在此模式下，你可以指定数据表中的每个点的值。
例如，要得到与绘图模式同样的弹簧特性曲线，你需要输入25个点的X-Y数据。

生成特性文件

- 在标准建模模式或模板建模模式下，要生成一个新的特性文件，可以在菜单 Tools 下选择Curve Manager。
- 要设置曲线编辑器到一个合适的模式下，选择一个新文件并指定你要生成什么类型的特性文件即可。
- 不同类型的特性文件：
 - ◆ 衬套(**Bushing**) – 指定定义三个方向移动和转动的刚度曲线，与 ADAMS/Solver中的 BUSHING 不同之处在于，这些曲线可以是非线性的。也可以包括阻尼系数在内。
 - ◆ 止挡(**Bumpstop**) – 一条刚度曲线(力—变形量)
 - ◆ 反向止挡(**Reboundstop**) -一条刚度曲线(力—变形量)
 - ◆ 弹簧(**Spring**) -一条刚度曲线(力—变形量) 和自由长度(有关弹簧的信息，参见第 10 章的弹簧部分)

生成特性文件

- ◆ 阻尼器(Damper) – 一条阻尼系数曲线 (时间—速度)
- ◆ 车轮包络空间(Wheel envelope) – 输入边界：
 - 转向输入 (长度或角度)
 - 车轮内部点和边界
 - 转向内部点和边界

注意：在曲线编辑器中不支持线形弹簧、阻尼器和衬套的特性文件。

- 曲线编辑器的功能：
 - ◆ 在窗口内全屏显示曲线
 - ◆ 显示曲线的局部
 - ◆ 曲线的数学运算
 - ◆ 垂直方向的热点
 - ◆ 记忆曲线的切换

生成特性文件

■ 曲线的数学运算

- ◆ 斜线(**Slope**) – 指定斜率、限界和数据点数或段数(等于数据点数 - 1)。
- ◆ 偏置(**Offset**) – 按照给定距离偏置曲线。
- ◆ 取绝对值(**Absolute value**) – 无参数，取曲线的绝对值。
- ◆ 取反(**Negate**) – 逆转曲线。
- ◆ 零起点(**Zero**) -偏置曲线，使其起点为 0。
- ◆ Y 数据镜像(**Y mirror**) – 相对于 y 轴镜像。
- ◆ X Y 数据镜像(**XY mirror**) -镜像 X 和 Y数据，曲线可以得到相同的数值。
- ◆ 拷贝X → Y(**Copy x → y**) – 使曲线的 y 值与 x 值相同。
- ◆ 函数(**Function**) – 指定一个函数，边界和数据点数或段数，你将得到所指定函数的曲线。

生成特性文件

- ◆ **插值(Interpolate)** – 使用下述的一个差值方法，并生成你所指定数据点数：

Akima	Cspline
Linear	Notaknot
Cubic	Hermite


- ◆ **过渡(Step)** – 指定起始点和结束点的 X 值以及对应起始点和结束点的 Y 值。
 - ◆ **缩放(Scale)** – 按照所给比例缩放曲线。
 - ◆ **斜坡(Ramp)** -指定起始点和结束点的 X 值以及对应起始点和结束点的 Y 值。
 - ◆ **扩展(Expand)** – 指定 X 的起始点和结束点的值。
 - ◆ **正弦(Sine)** -起始点和结束点的 X 和 Y 的值，何时开始起振，最小、最大幅值、频率，数据点数或段数。
- 注意：**对于车轮包络空间，数学运算无效。

特性文件编辑器

- 特性文件编辑器允许你编辑元件数据并以 XML 格式存储。下列元件使用新的对话框格式：
 - ◆ 空气弹簧(Air springs)
 - ◆ 止挡(Bumpstops)
 - ◆ 衬套(Bushings)
 - ◆ 阻尼器(Dampers)
 - ◆ 反向止挡(Reboundstops)
 - ◆ 弹簧(Springs)
- 参见共享数据库下的 XML 格式特性文件的例子。

特性文件编辑器

■ 要打开特性文件编辑器：

1. 进入 ADAMS/Car 标准界面。
2. 打开一个子系统。
3. 在一个衬套上点击鼠标右键并选择 **Modify**。
4. 在 **Property File** 文本栏点击鼠标右键并选择 **acar_shared** 数据库下的一个 **.xml** 文件。
5. 选择  并存储你的修改。

转换 TeimOrbit 格式文件为 XML格式文件

- 此版本的新功能：在模板中，XML 格式特性文件为缺省选项，在修改时显示特性文件编辑器，你可以修改数据。
- 你可以使用新版功能，转换或者单个文件，或者整个数据库，从TeimOrbit 格式到 XML格式 (Tools → Database Management → Version Upgrade → TeimOrbit →XML)。

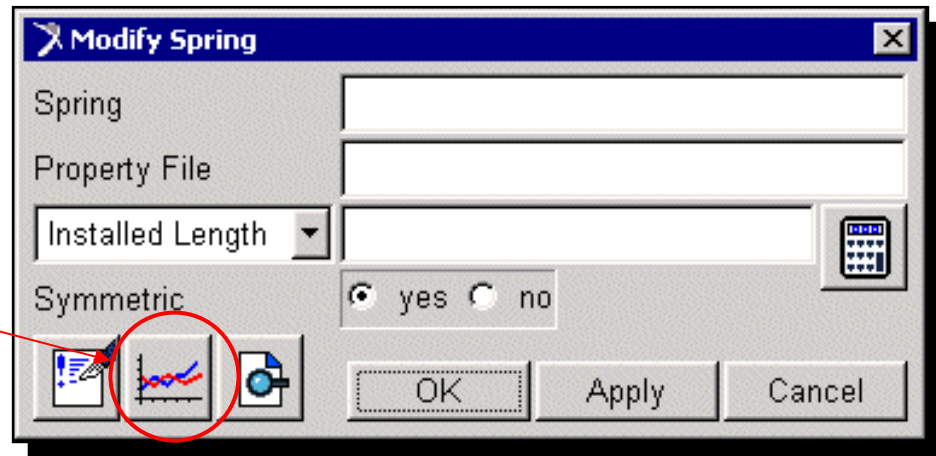
转换 TeimOrbit 格式文件为 XML 格式文件

- 在转换整个数据库时，软件接受你想要转换的数据库、文件类型以及目标可写数据库，读入所有需要转换的文件再转换，最后存储到目标数据库下。为防止出现在 XML 格式文件可以使用而在原来 TeimOrbit 格式文件未定义或不支持的情况发生，软件使用一系列的缺省设置。
- 转换完成后，你可以在特性文件编辑器中打开 XML 格式的文件了。

修改现有的特性文件

- 要修改现有的特性文件，你可以：
 - ◆ 当你想要生成一个特性文件时，你可以在 ADAMS/Car 标准界面或模板界面下使用菜单 Tools 打开曲线编辑器(Curve Manager)。此处，你将打开一个特性文件编辑修改并使之生效。
 - ◆ 使用修改对话框打开曲线编辑器和特性文件编辑器(只对.xml 文件)，所选择的曲线将被自动的打开以便你进行编辑。

曲线编辑器和
特性文件编辑
器(针对 .xml格式)

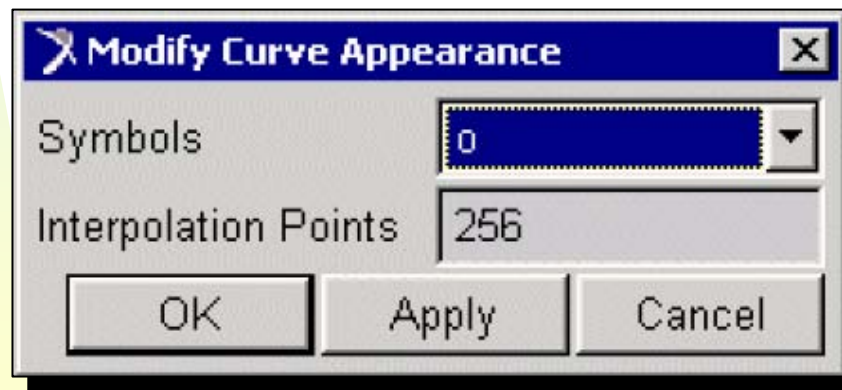


绘图 vs. 数表

- 要在绘图和数表之间转换，在曲线编辑器下的主菜单下选择 View 命令并选择 Plot 或 Table，关键看你想进入什么模式。你只能从绘图模式下关闭曲线编辑器，退回到标准模式或模板建模模式。如果你是在数表模式，先进入绘图模式再选择 File → Close。
- 当你完成特性文件的编辑后，你可以存储该文件。ADAMS/Car 存储到缺省数据库的对应表目录下。例如，ADAMS/Car 存储一个弹簧的特性文件到数据库的表目录 spring.tbl 下。

曲线 vs. 数表

- 要查看缺省的数据库是什么：
 - ◆ 从菜单 **Tools** 下指向 **Database Management**，再选择 **Database Info**。
 - ◆ 看一下绘图的子标题，表示特性文件的完整路径。
- 在曲线编辑器下，要改变表示数据点的符号，可以从菜单 **Settings** 下选择 **Appearance**。



练习 4 使用曲线编辑器修改弹簧特性

- 在本练习中，你将使用曲线编辑器修改一个悬架分析下的弹簧的特性文件。



本练习大约需要半个小时完成。

练习 4 使用曲线编辑器修改弹簧特性

■ 设置模型

要设置模型：





1. 在 ADAMS/Car 标准界面下打开 **mdi_front_vehicle** 装配。
2. 运行一次反向双轮跳分析：
 - **Output Prefix: baseline**
 - **Number of Steps: 10**
 - **Bump Travel: 100**
 - **Rebound Travel: -100**

Note: 上下跳动的距离的单位为缺省的单位 mm。

练习 4 使用曲线编辑器修改弹簧特性

■ 修改弹簧

要修改弹簧：

1. 在任何一个弹簧上点击鼠标右键并选择 **Modify**。
2. 选择 **Curve Manager** 工具按钮 。
3. 点击当前曲线的记忆曲线的可视性切换按钮 。
4. 右键点击 **Curve Math** 工具包 ，再选择 **Scale** 按钮 。
5. 在 **Scale Value** 文本栏内，输入 **1.5**。
6. 选择 **Apply**。

Curve Manager 亮显新的曲线，与原曲线相比，放大了1.5倍(虚线)。

练习 4 使用曲线编辑器修改弹簧特性

■ 存储弹簧特性文件

要存储弹簧特性文件：

1. 从菜单 **File** 下选择 **Save As**。
2. 命名新文件 **my_spring.spr**。
3. 选择 **OK**。

ADAMS/Car 存储新文件在你的缺省写数据库下。

4. 从菜单 **File** 下选择 **Close**。

ADAMS/Car 显示一个对话框，询问你是否要在你的模型中使用此弹簧特性文件。

5. 选择 **Yes**。
6. 在 **Modify Spring** 对话框内选择 **OK**。

练习 4 使用曲线编辑器修改弹簧特性

■ 运行一次仿真

- ◆ 运行一次仿真并在 ADAMS/PostProcessor 中比较结果。

■ 要运行一次仿真：

1. 运行一次反向双轮跳分析，同上一次名为baseline分析一样，只是在 **Output Prefix** 中输入 **new_spring**。
2. 要启动 ADAMS/PostProcessor，从菜单 **Review** 下选择 **Postprocessing Window** 或按 **F8** 键。
比较 baseline 与 new_spring 的仿真结果，确认其是否有所区别(例如，看看车轮速率)。
3. 要返回标准界面，从菜单 **File** 下选择 **Close Plot Window** 或按 **F8**。

练习 4 使用曲线编辑器修改弹簧特性

- 转换为 XML 格式文件
- 要转换TeimOrbit文件my_spring.spr 为 XML格式文件：
 1. 从菜单 **Tools** 下指向 **Database Management** , 再指向 **Version Upgrade** , 然后再选择 **TeimOrbit → XML**。
 2. 设置 **Convert** 为 **Single File**。
 3. 设置 **File Type** 为 **spring**。
 4. 在 **Property File** 文本栏内点击鼠标右键并选择 **<acar_training>/springs.tbl/ my_spring.spr**。
 5. 设置 **To Database** 为 **acar_training**。
 6. 选择 **OK**。

练习 4 使用曲线编辑器修改弹簧特性

- 替换 TeimOrbit 格式特性文件
- 要使用一个 XML 文件替换 TeimOrbit 格式特性文件
 1. 在弹簧上点击鼠标右键并选择 **Modify**。
 2. 在 **Property File** 文本栏内点击鼠标右键并选择
`<acar_training>/springs.tbl/ my_spring_spr.xml`。

练习 4 使用曲线编辑器修改弹簧特性

- 修改 XML 特性文件

- 要修改 XML 特性文件：

1. 选择 **Curve Manager/Property File Editor** 工具 。
出现 Property File Editor。
2. 在 **Properties** 表内选择 **Plot/Edit Data**，定位到 **Spline Data**的右侧。
你会看到与 Curve Manager 内同样的数据。你可以牵引曲线上的热点或编辑数表值，就像在曲线编辑器中一样。同时，注意输入输出数据选项，输入一个一般的函数以定义你的特性文件，控制曲线图的视窗大小等功能。
3. 计算非线性弹簧在 20 mm处的弹簧刚度(提示： $1500/20 = 75 \text{ N/mm}$)。

练习 4 使用曲线编辑器修改弹簧特性

4. 熟悉一下这些工具，并选择 **Cancel** 退出 Curve Editor，不保存你的修改。
5. 改变 **Method** 从 **Nonlinear** 为 **Linear**。
6. 选择 **Header** 表并标注单位从 **Length** 为 **Force**。
7. 选择 **Properties** 表并输入 **75** 作为线形比率。
8. 选择 **Save As** 并选择 **OK** 以存储为 **my_spring_spr.xml**。
不像 TeimOrbit 文件，XML 格式特性文件存储此圆弹簧的线型和非线性的特征值，并允许你在不同的类型间切换。
9. 要替换已有的文件，选择 **Yes**。
10. 要关闭 Property File Editor，选择 **Exit**。
11. 要接受新的弹簧特性文件，选择 **OK**。

练习 4 使用曲线编辑器修改弹簧特性

■ 运行一次仿真

1. 运行一次反向双轮跳分析，同上一次名为my_spring分析一样，只是在 **Output Prefix** 中输入 **linear**。
2. 启动 ADAMS/PostProcessor。
3. 比较 仿真结果，确认其是否有所区别(例如，看看车轮速率)。如果你还有时间的话，可以试试 Curve Manager/Property File Editor中其它的功能。

第 5 章 输出配置文件

- 本章主要介绍输出配置文件，如何生成输出配置文件，如何观察你的分析结果。

绘图配置文件

- 本章内容：
 - ◆ 绘图配置文件
 - ◆ 生成绘图配置文件
 - ◆ 利用绘图配置文件生成绘图
 - ◆ 绘图配置文件的格式
 - ◆ 绘图配置文件的例子

绘图配置文件

- 绘图配置文件定制你的基于模板建模产品的后处理配置情况：
 - ◆ 生成什么曲线
 - ◆ 曲线图的垂直坐标轴和水平坐标轴的成分
 - ◆ 一般的设置和偏好，诸如：标题、标签、纵或横轴的间距、比例、图例等。
- 此文件目前只支持每页面一个曲线图，可以包含多个坐标轴。你可以使用同一个绘图配置文件对应多次仿真结果。
- 绘图配置文件是 TeimOrbit 格式文件并存储在你的数据库 plot_configs.tbl 目录下。
- 你可以在 ADAMS/PostProcessor 中使用绘图配置文件的功能。

生成绘图配置文件

- 你可以生成当前在ADAMS/PostProcessor 中所有的绘图页或部分绘图页的绘图配置文件。你所使用的基于模板建模产品存储绘图配置文件在你的缺省写数据库下的 plot_config 表目录下。
- 注意你每页只能有一个绘图。

生成绘图配置文件

■ 要生成绘图配置文件：

1. 根据需要生成并设置绘图，包括指定标签、间距等等。比如，你可以生成一系列的绘图，添加子标题以便描述绘图有关的分析的类型等。
2. 从菜单 **File** 下指向 **Export**，再选择 **Plot Configuration File**。
出现 Save Plot Configuration File 对话框。
3. 在 **Configuration File Name** 文本栏内输入生成绘图配置文件的名字。
4. 选择你想涵盖的绘图和曲线。
5. 选择 **OK**。

利用绘图配置文件生成绘图

- 在你运行一次仿真后，你能够看到由绘图配置文件所定义的一系列的绘图。如果你的绘图配置文件包含用户化的命令关键字以生成绘图和曲线，你可以使基于模板的产品激活包含命令关键字的宏。
- 绘图配置文件指定你的绘图的子标题。另外，在 File Import 对话框，你可以：
 - ◆ 在所有的绘图页面上添加标题。
 - ◆ 在一个绘图页面上通过交叉绘图选项绘制多次分析结果的曲线。
 - ◆ 改变你的绘图页面的外观，比如字体及大小，使用可选的可执行的用户化宏。要使用这一选项，你必须首先定义一个包含要执行命令的宏。

利用绘图配置文件生成绘图

■ 要观察在绘图配置文件中定义的绘图：

1. 从菜单 **Plot** 下选择 **Create Plots**。
出现 File Import 的对话框。
2. 在 **Analyses** 文本栏内输入分析结果的名字或你想观察结果的分析的名字。
3. 在 **Plot Configuration File** 文本栏内输入绘图配置文件的名字，此绘图配置文件定义你要观察的绘图。
4. 选择 **OK**。



更详细节，参见 ADAMS/Car 在线帮助的 Plot tab 部分。

绘图配置文件的格式

■ 绘图数据块

◆ 绘图数据块的结构如下：

- NAME
- SUBTITLE
- TIME_LOWER_LIMIT
- TIME_UPPER_LIMIT
- PLOT_AXES (subblock)
- COMMAND
- command_keyword

◆ 如果你定义了下面的命令关键字，基于模板建模产品在生成每个绘图后会执行这些命令。

acar custom_plots <command_keyword> &
plot_name=<plot_name>

- ◆ 命令 acar custom_plots <command_keyword> 必须已存在当前进程下，或交互式地或直接存在在 acar.bin、aengine.bin 或 arail.bin 下。
- ◆ 更多信息，参见 KBA 8848：

<http://support.adams.com/kb/faq.asp?ID=kb8848.html>.

绘图配置文件的格式

■ 绘制曲线数据块

◆ 绘制曲线数据块的结构如下：

- NAME——名称
- PLOT——曲线
- VERTICAL_AXIS——纵坐标轴
- HORIZONTAL_AXIS——横坐标轴
- VERTICAL_COMPONENT——纵向数据
- HORIZONTAL_COMPONENT——横向数据
- LEGEND_TEXT——图例
- COLOR——颜色
 - red, blue, yellow, magenta, cyan, black, white, skyblue, midnight_blue, blue_gray, dark_gray, silver, peach, maize
- STYLE——线型
 - solid, dash, dotdash, dot
- SYMBOL——符号
 - none, x, o, plus, star, at
- LINE_WEIGHT——粗细
 - Real value from 1-4
- COMMAND——命令
 - command_keyword

绘图配置文件的格式

- ◆ 如果你定义了下面的命令关键字，基于模板建模产品在生成每个绘图中的曲线后会执行这些命令。

```
acar custom_plots <command_keyword> &  
analysis=<analysis> &  
plot_name=<plot_name> &  
vertical_data=<y> &  
horizontal_data=<x> &  
curve_name=<curve_name>
```

- ◆ 命令 `acar custom_plots <command_keyword>` 必须已存在当前进程下，或交互式地或直接存在在 `acar.bin`、`aengine.bin` 或 `arail.bin` 下。
- ◆ 更多信息，参见 KBA 8848:
<http://support.adams.com/kb/faq.asp?ID=kb8848.html>.
- ◆ 我们在 ADAMS/Car 的共享数据库下提供几个绘图配置文件的例子。
更多信息，参见 KBA 10652:
<http://support.adams.com/kb/faq.asp?ID=kb10652.html>.

绘图配置文件的例子

- 右图为一个ADAMS/Car 的绘图配置文件的例子：

```
$-----plot
[PLOT]
NAME = 'toe_angle'
SUBTITLE = 'Toe Angle'
TIME_LOWER_LIMIT = 1
TIME_UPPER_LIMIT = 1e+10
(PLOT_AXES)
{axis_name type label scaling divisions lower_limit upper_limit}
'vaxis' 'vertical' 'Angle' 'linear' 6 0 0
'haxis' 'horizontal' 'Wheel Travel' 'linear' 10 0
$-----plot_curve
[PLOT_CURVE]
NAME = 'toe_angle_r'
PLOT = 'toe_angle'
VERTICAL_AXIS = 'vaxis'
HORIZONTAL_AXIS = 'haxis'
VERTICAL_COMPONENT = 'toe_angle.right'
HORIZONTAL_COMPONENT = 'wheel_travel.vertical_right'
LEGEND_TEXT = 'Right'
COLOR = 'red' STYLE = 'solid'
SYMBOL = 'none'
LINE_WEIGHT = 1.5
$-----plot_curve
[PLOT_CURVE]
NAME = 'toe_angle_l'
PLOT = 'toe_angle'
VERTICAL_AXIS = 'vaxis'
HORIZONTAL_AXIS = 'haxis'
VERTICAL_COMPONENT = 'toe_angle.left'
HORIZONTAL_COMPONENT = 'wheel_travel.vertical_left'
LEGEND_TEXT = 'Left'
COLOR = 'blue' STYLE = 'dash'
SYMBOL = 'none'
LINE_WEIGHT = 2.0
```

练习 5 - 生成绘图配置文件

■ 问题阐述

- ◆ 本练习中，简单的生成一系列的悬架绘图，并存储练习 5 - 生成绘图配置文件。



本练习大约需要半个小时完成。

练习 5 - 生成绘图配置文件

■ 生成绘图配置文件的基本步骤：

1. 在标准界面下运行一个悬架分析。
2. 手工在 ADAMS/PostProcessor 中绘制若干个曲线图。
3. 在 ADAMS/PostProcessor 中存储绘图配置文件 (.plt)。
4. 在标准界面下再次运行分析。
5. 在 ADAMS/PostProcessor 中加载绘图配置文件 (.plt)。你会自动地得到同样的曲线图。

第 6 章 生成并仿真悬架系统

- 在本章中，你将学习如何在 ADAMS/Car 中生成一个悬架装配。同时，你还将了解在 ADAMS/Car 中可以进行的悬架分析以及如何提交计算。

生成并仿真悬架系统

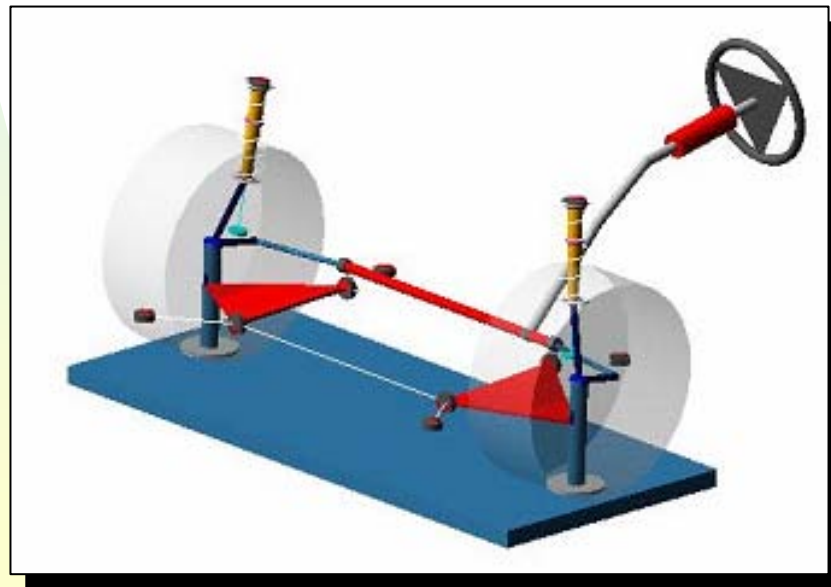
- 本章内容：
 - ◆ 生成悬架装配
 - ◆ 二分之一车辆分析
 - ◆ 悬架参数
 - ◆ 生成载荷工况
 - ◆ 警告信息
 - ◆ 分析所对应的结果文件

生成悬架装配

- ◆ 一个装配包含一个试验台和一个或多个子系统 (试验台可以理解为就是一个特殊的子系统)。
- ◆ 你需要在标准界面下生成悬架的装配：从菜单 File 下指向 New，然后选择 Suspension Assembly。在对话框中，指定装配中所有的子系统，以及试验台。
- ◆ 如果你使用由新的模板所生成的子系统，你需要确认信息交换器的匹配关系。信息交换器为在模板建模模式下定义的特殊变量，以使子系统之间或与试验台之间相互传递一些有关装配的信息，保证使其能够正确的装配在一起。在模板建模模式下，你可以测试信息交换器是否能与其它模板匹配 (信息交换器将在第 11 章—信息交换器 部分详细阐述)。在生成装配时，ADAMS/Car 将在信息窗口内显示不匹配的信息交换器，供你参考调试模型。

生成悬架装配

- 下图为一个悬架的装配，包含一个前悬架子系统、一个转向子系统和试验台。
- 因为仿真过程是由试验台所激励的，要进行仿真，你必须使用装配模型。



二分之一车辆分析

- 在 ADAMS/Car 中，你可以进行下列类型的悬架分析：
 - ◆ **Parallel wheel travel** – 两个车轮同向跳动。
 - ◆ **Opposite wheel travel** -两个车轮反向跳动。
 - ◆ **Roll and vertical force** – 作用在两个车轮上的合力保持恒定情况下，扫描转向角，找到其垂向的位置。
 - ◆ **Single wheel travel** – 一个车轮固定，另一个车轮运动。
 - ◆ **Steering** – 在方向盘上或转向拉杆上施加运动驱动。

二分之一车辆分析

- ◆ **Static load** – 在指定位置(车轮中心或轮胎着地点)施加载荷或位移，可以考虑，也可以不考虑转向
- ◆ **External files:**
 - **Loadcase** - Essentially a selection of previous events.
 - **Wheel envelope** – 车轮同向跳动同时施以转向，以得到车轮所扫过的空间的体积。
- ◆ 当进行悬架分析时，ADAMS/Car 在前一秒内先将车轮的中心调到最低的位置，然后在你给定的时间、步数内将悬架驱动到最高位置。



更多信息，参见 ADAMS/Car 的在线帮助 Analyze tab 部分。

悬架参数

- 一些缺省的输出要求所需要的信息并未包含在模型中，因此，你必须提供这些附加信息。这些信息不在仿真的输出中，因为它们只影响一些用户定义的结果(例如，滚转中心等)。
- ADAMS/Car 以数组的形式存储这些输入项，名为悬架参数，你可以在标准界面下选择 Simulate → Suspension Analysis → Suspension Parameters。
- 你必须提供的值有：
 - ◆ Loaded tire radius

悬架参数

- ◆ Tire stiffness (用户输入或来自特性文件)
- ◆ Sprung mass
- ◆ CG height
- ◆ Wheelbase
- ◆ Drive ratio
- ◆ Brake ratio
- ◆ ADAMS/Car 使用参数 cg_height、wheelbase、sprung_mass和 tire stiffness计算：
 - Percent anti-dive
 - Dive
 - Percent anti-lift
 - Lift
 - Percent anti-squat
 - Roll center height
- ◆ ADAMS/Car 使用 tire stiffness 计算：
 - Suspension rates
 - Total roll rate

生成载荷工况

- 载荷工况文件为 ASCII 格式文件，包含运行仿真所有必须的信息。基本上讲是进行脚本式的悬架分析方式，包含下列五种分析类型：
 - ◆ Parallel, opposite and single, wheel travel
 - ◆ Steering
 - ◆ Static load
 - ◆ Roll and vertical force
- 当运行一个载荷工况分析时，ADAMS/Car 搜索存储在数据库下特定的载荷工况文件。你可以调用多个载荷工况文件，ADAMS/Car 将一次全部运行。
- 你可以生成一个载荷工况文件，通过选择 Simulate → Suspension Analysis → Create Loadcase。然后选择你要进行仿真的类型并指定相关的数据。

警告信息

- ◆ 当你生成一个装配时，你有时会看到一些警告信息。例如，你生成一个没有转向子系统或车身子系统的前 MacPherson 式悬架装配时。因为转向和车身部件没有在试验台或其它子系统中被指定。这样的话，某些信息交换器将连接到大地上，或根本就不连接到任何对象上。显示的信息窗口如下图所示：

```
Creating the suspension assembly: 'macph_assy'...
Moving the rear suspension subsystem: 'my_macph'...
Assembling subsystems...
Assigning communicators...
WARNING: The following input communicators were not assigned during assembly:
  my_macph.cil_tierod_to_steering (attached to ground)
  my_macph.cir_tierod_to_steering (attached to ground)
  my_macph.cis_subframe_to_body (attached to ground)
  my_macph.cil_strut_to_body (attached to ground)
  my_macph.cir_strut_to_body (attached to ground)
  my_macph.cil_ARB_pickup
  my_macph.cir_ARB_pickup
  testrig.cis_steering_wheel_joint
  testrig.cis_steering_rack_joint
  testrig.cis_leaf_adjustment_steps
  testrig.cis_powertrain_to_body (attached to ground)
Assignment of communicators completed.
Assembly of subsystems completed.
Suspension assembly ready.
```

警告信息

- 在汽车模块下的共享数据库中的模型都包含所有可能被其它子系统所调用的信息交换器，并且在很多情况下并不是所有的信息交换器都需要被使用。然而，一定要检查哪些没有被连接使用，并看看是否有没有关系。此处，它们中的大部分可能连接到车身上或其它子系统，但你现在所做的分析中却并不关心。缺省的情况下，如果 ADAMS/Car 不能进行信息交换器匹配的话，它将连接到大地上，这种情况是最好的。因此，通常情况下，你无须关心信息交换器，除非你看到这样的提示：a communicator that should be used, but isn't，即一个信息交换器应该被使用，但是没有。这时，你需要检查你的模型了。

分析所对应的结果文件

- ◆ 很重要一点需要知道，所有的专业产品，包括 ADAMS/Car 在内，都只是 ADAMS/Solver 的前后处理。意味着，它们只是生成 .adm 文件 (ADAMS/Solver 数据文件，模型文件) 和 .acf 文件 (ADAMS/Solver 命令)。这些文件存放在工作目录下 (File → Select Directory)。
- ◆ 输出文件包括：信息文件(.msg)、测试文件(.req)、结果文件(.res)、图像文件(.gra) 和输出文件(.out)。ADAMS/Car 有可能还输出 .dcf 文件和 .dcd 文件，这部分将在第 8 章 驾驶机器中阐述。注意，如果发现模型中包含弹性体，ADAMS/Car 将自动存储结果文件(.res)。因为 ADAMS/PostProcessor 需要该结果文件(.res) 以绘制弹性体变形颜色等等。
- ◆ 当你在 ADAMS/PostProcessor 导入分析测试文件(.req) 时，ADAMS/Car 同时要产生一个特殊的文件，叫名字文件 (.nam)，其中包含在界面中与每个测试请求相关联的名字。
- ◆ 你可以使用 .adm、.acf 和 .nam 文件直接在 ADAMS/Car 的图形界面外提交 ADAMS/Solver 解算。

练习 6 – 运行悬架分析

- 完成 *Getting Started Using ADAMS/Car* 的第11页上的悬架分析。
- 在练习的第 27 页，练习者需要选择 ‘steering_wheel_input’。这个项在进入后处理 ADAMS/PostProcessor 的 ‘+testrig’ 的下一层。
- 类似地，在第 29 页，要绘制 ‘scrub_radius’ 时，也是在 ‘+testrig’ 的下一层。



本练习大约需要半个小时完成。

第 7 章 生成并仿真整车模型

- 本章，你将学习如何在 ADAMS/Car 中生成整车装配。你还将学习在整车装配分析中可以进行的分析类型以及如何提交解算。

生成并仿真整车模型

■ 本章内容：

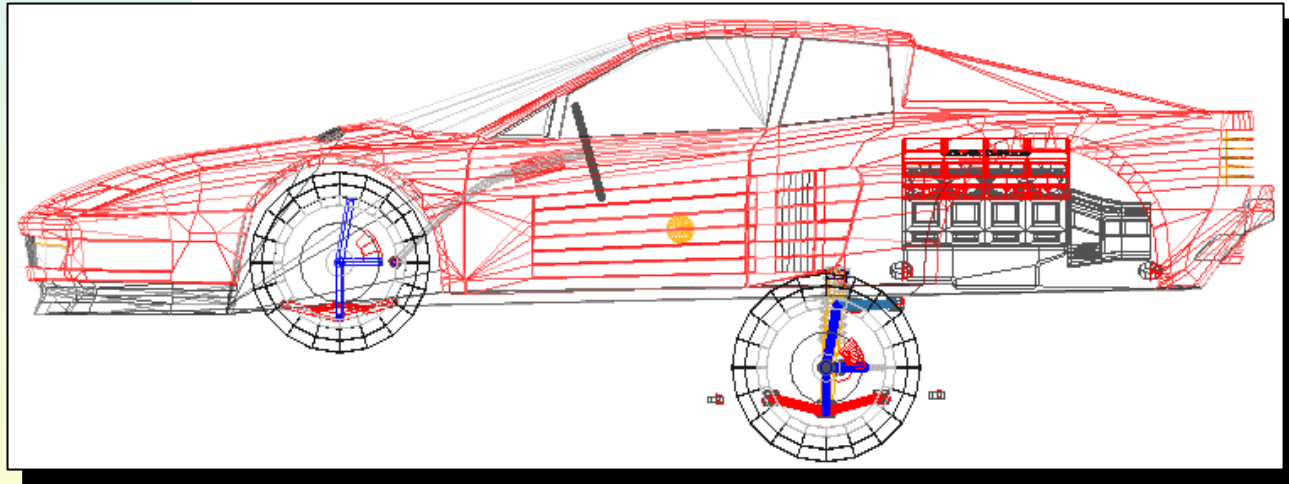
- ◆ 生成整车装配
- ◆ 移动子系统的装配位置
- ◆ 更新子系统和装配
- ◆ 使子系统和装配同步
- ◆ 增加/减少和激活/失效子系统
- ◆ 调整质量特性
- ◆ 整车分析
- ◆ 整车分析事件生成器

生成整车装配

- ◆ 要生成整车装配，从 File → New → Full Vehicle。需要在此对话框内填上一系列组装装配的子系统。需要的子系统包括车身子系统、前后悬架子系统、前后轮胎子系统、转向子系统和试验台。
- ◆ 现在所有可以使用的分析都是基于驾驶机器。因此，要完成开环、闭环或准静态分析，在装配模型时，你必须选择选择试验台 __MDI_SDI_TESTRIG。
- ◆ 要在整车装配中包含其它子系统，可以选择 Other Subsystem。这些子系统在装配中的位置和连接关系取决于子系统和信息交换器是如何定义以及模板是否进行位置调整等。

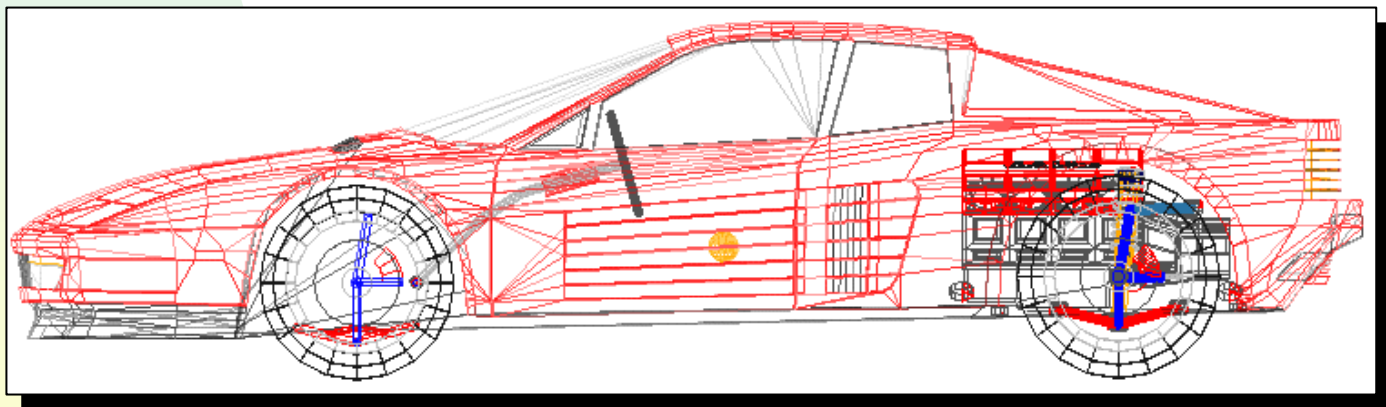
移动子系统的装配位置

- 要在标准界面移动子系统的装配位置，可以从Adjust → Shift。现在你可以将子系统前/后或上/下移动其装配位置。前/后移动为将子系统沿绝对坐标系的 X-轴移动。上/下移动为将子系统沿绝对坐标系的 X-轴移动。



移动子系统的装配位置

将后悬架系统上/下移动后



更新子系统和装配

■ 更新子系统

- ◆ 偶尔地，你可能想要改变你现有装配中的子系统，并使改变其作用。也就是，你想要编辑子系统文件并反映到你的装配文件。要实现这一想法，可以 File → Update → Subsystem，再选择子系统。这样操作的好处在于你不用关闭整个装配再重新打开它以查看子系统的改变。

注意：你必须存储修改了的子系统在将要调入装配的数据库下。同样，更新子系统不会影响拓扑结构，而只能更新在子系统级可以更新的子系统参数。

■ 更新装配

- ◆ 与更新子系统相类似，你可以更新装配。从 File → Update → Assembly，再选择 assembly。这样操作的好处在于你不用关闭整个装配再重新打开它以再次加载已存子系统的信息。

使子系统和装配同步

- ◆ ADAMS/Car 允许你将同一进程中的同一子系统自动地保持同步。你能够选择一个主子系统(源)以及选定进程中其它子系统(目标)的动态更新列表。目标子系统将继承主子系统所有的设置(只有子系统模式被保留)，这样有助于减少某些子系统一致性方面增加的考虑，当我们在同一进程的多个装配中引入同样的子系统。
- ◆ 例如，假定你现在打开了一个悬架装配和一个整车装配，两个装配中都使用同一个转向子系统。如果你在悬架装配中修改了方向盘的位置，此功能可以使你在整车装配中保持同步，而不需要再次在整车装配中修改。

增加/减少和激活/失效子系统

一个装配为一个激活的实体，即使在装配打开后，你也可以修改其下属的子系统。在ADAMS/Car中，你可以对装配进行下列的操作，如下所述：

- 在现有装配中增加一个新的子系统：
 - ◆ File → Manage Assemblies → Add Subsystem
 - ◆ 装配首先被解开装配关系，包括不分配信息交换器。然后在现有的装配下打开新的子系统。最后，再行组装整个装配系统，包括再分配信息交换器。

增加/减少和激活/失效子系统

- 在装配中以新的子系统替换原来的子系统
 - ◆ File → Manage Assemblies → Replace Subsystem
 - ◆ 装配首先被解开装配关系，包括不分配信息交换器。然后从装配中删除原来的子系统。然后在现有的装配下打开新的子系统。最后，再行组装整个装配系统，包括再分配信息交换器。
- 从装配中删去一个子系统：
 - ◆ File → Manage Assemblies → Remove Subsystem
 - ◆ 装配首先被解开装配关系，包括不分配信息交换器。然后从装配中删除原来的子系统。最后，再行组装整个装配系统，包括再分配信息交换器。

增加/减少和激活/失效子系统

- 在一个装配中失效一个子系统
 - ◆ File → Manage Assemblies → Toggle Subsystem Activity
 - ◆ 装配首先被解开装配关系，包括不分配信息交换器。然后从装配中失效一个子系统，这并不意味着从装配中删除该子系统，而只是忽略而已。最后，再行组装整个装配系统（不考虑失效的子系统），包括再分配信息交换器。

增加/减少和激活/失效子系统

■ 在一个装配中激活一个子系统

- ◆ File → Manage Assemblies → Toggle Subsystem Activity
- ◆ 装配首先被解开装配关系，包括不分配信息交换器。然后从装配中激活一个现有的子系统，这意味着在装配中考虑该子系统。最后，再行组装整个装配系统（不考虑失效的子系统），包括再分配信息交换器。
- ◆ 在这个版本之前，要完成这些操作，需要先关闭装配，再重新打开新的装配，新的装配中再考虑包括哪些新的子系统。

调整质量特性

■ 自动地调整车辆的质量特性

- ◆ 在 ADAMS/Car 中，您能够一个装配模型的质量特性。要调整总质量，可以输入想要的质量、主惯量以及相对于一个标记点的质心位置等，你还要选择一个ADAMS/Car要调整质量特性的部件，以使整个装配与你的实际的车辆的质量特性相匹配。
- ◆ 要调整车辆的质量特性，可以从 Simulation → Full Vehicle Analysis → Automatic Mass Adjustment.
- ◆ 要取得模型中部分或所有的部件的质量总和，可以从 Tools → Aggregate Mass (可以使用对话框下的帮助键 (F1) 得到有关合质量 Aggregate Mass方面的帮助)。
- ◆ 如果按照你给定的数据输入，导致你所选择的部件的质量特性中出现负的 I_{xx} 、 I_{yy} 或 I_{zz} ，表明系统是不稳定的。为避免出现这个问题，你应该先减小其它部件的转动惯量的大小，使得所有的部件的转动惯量满足 $I_{xx} + I_{yy} > I_{zz}$ ，而不出现负的转动惯量。

调整质量特性

■ 根据典型的测试调整车辆的质量特性

- ◆ 在大多数情况下，你手头有的车辆重心(CG)的位置数据(惯性张量)是相对于车辆重心(CG)的。然而，当你使用 Tools → Aggregate Mass 所得到的惯性张量是相对于在(0,0,0)处大地坐标系的，因为以前你并不知道车辆重心(CG)的位置。所以，你需要首先找到车辆重心(CG)的位置，然后在模板里相应的位置处，建立一个所谓的 CG 标记点，然后再使用 Tools → Aggregate Mass，这次相对于 CG，再次找到惯性张量的大小。以此方法，可以将你的 MSC.ADAMS 模型的质量特性与你实测的数据进行比较。

调整质量特性

- ◆ 在你 ADAMS/Car 的车身模板中，你可以生成一个名为 vehicle_cg 的硬点，再生成一个名字也为 vehicle_cg 的标记点，其位置与硬点相同。在 ADAMS/Car 标准界面下，首先调整硬点 CG 的位置为指定值，然后选择 Simulate → Full-Vehicle Analysis → Automatic mass adjustment。要显示此对话框的帮助，按 F1 键。此处为对话框内建议输入的数据：
 - CG location = 0,0,0
 - Relative to marker = vehicle_cg
 - Modify Part = body (chassis)

ADAMS/Car 通过调整被修改部件的质量、转动惯量和质心位置达到修改整车的质量特性的目的。

提示：如果你选择 ges_chassis 为被修改部件，不要设置 Relative to Marker 为 ges_chassis.cm，ADAMS/Car 将改变被修改部件的质心位置。

整车分析

- 正如前面所指出的，目前所有可以调用的分析都是基于驾驶机器。因此，要进行开环的、闭环的或准静态分析，你必须选择 `._MDI_SDI_TESTRIG` 在你的装配中。
- 下面为在 ADAMS/Car 中可以进行的分析列表：
 - ◆ 开环分析
 - Drift
 - Impulse steer
 - Ramp steer
 - Single lane change
 - Step steer
 - Swept sine steer

整车分析

- ◆ 转向分析
 - Braking-in-turn
 - Constant-radius cornering
 - Cornering with steering release
 - Lift-off turn-in
 - Power-off cornering
- ◆ 直线特性
 - Acceleration
 - Braking
 - Power-off straight-line
- ◆ 移线分析
 - ISO lane change
- ◆ Dcf 驱动

整车分析

- ◆ 准静态操纵性分析
 - Constant-radius cornering
 - Constant-velocity cornering
 - Force moment method
 - Straight-line acceleration
- ◆ ADAMS/Driver
- ◆ 3D smooth road

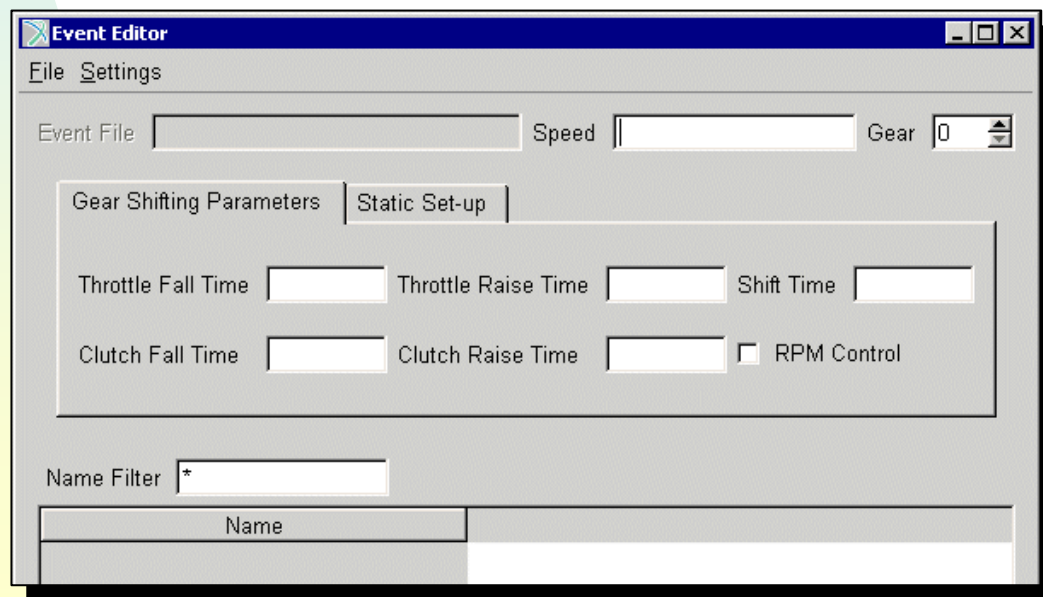


有关这些分析的细节，参见 ADAMS/Car 在线帮助的 Analyze 部分。

- ◆ 你可以进行其它整车的分析，但需要生成你自己的试验台 (参见附录 B：四轮垂向激励试验台) 或使用 ADAMS/Car Ride 和 ADAMS/Driveline (参见第 16 章 – 其它应用)。

整车分析事件生成器

- ◆ 你可以使用 Event Generator 以生成和修改整车分析事件文件(full-vehicle event files)，整车分析事件文件中包含你定义的多条mini-maneuvers。在每个 mini-maneuver内，你能够指定转向(steering)、油门(throttle)、制动(braking)、档位(gear)和离合器(clutch)的设置，你可以从一系列的终止条件中选择一个以结束每个 mini-maneuver。这些事件文件随后可以被MSC.Software开发的新驾驶技术所使用，以便进行各种整车的分析。新驾驶技术正在替代原来系列闭环控制仿真，就是熟知的 DriverLite。



整车分析事件生成器

- ◆ 驾驶机器控制文件使用 XML 格式的数据文件，可以在 Event Builder 界面下察看或其它任何的 Web 浏览器下打开。
- ◆ 要打开事件生成器，从 **Simulate → Full-Vehicle Analysis → Event Generator**。使用事件生成器定义通用的整车仿真事件或修改一个已有的整车仿真事件。你能设置：
 - 定义驾驶机器在一个事件执行过程中控制车辆所使用的车辆特定的某些信息。
 - 定义控制参数和换档策略。
 - 定义准静态设置信息，用于在仿真分析的预备阶段的设置，以使车辆达到所要求的纵向和横向加速度状态。
 - 定义基本操控的数目和类型；对每个基本操控，你可以指定：
 - 所有的五项操控指标的行为 (油门、刹车、档位、离合及转向)。
 - 一系列单个的或组合的结束条件。
 - 车辆纵向和横向动力学的机器控制策略。

练习 7 – 进行整车分析

- 完成 *Getting Started Using ADAMS/Car* 第 107 页整车分析练习。



本练习大约需要半个小时完成。

第 8 章 驾驶机器

- 驾驶机器根据你的指示驱动你的虚拟车辆，就像一名试车驾驶员驾驶一辆真车一样。

驾驶机器

■ 本章内容：

- ◆ 标准驾驶员界面 (SDI) 和驾驶机器
- ◆ 为什么使用 SDI?
- ◆ 数据流程
- ◆ 使用驾驶机器
- ◆ 限制
- ◆ 生成 SDI 的输入
- ◆ 驾驶员控制文件(Driver Control Files)
- ◆ 驾驶员控制文件(.dcf)的例子
- ◆ 生成 .dcf 和 .dcd 文件
- ◆ 驾驶员控制文件(.dcf)的结构
- ◆ 指定结束条件
- ◆ 可扩展的结束和退出条件
- ◆ 指定属性
- ◆ 驾驶员控制数据文件(Driver Control Data Files)
- ◆ 驾驶员控制数据文件(Driver Control Data Files)的构造体系
- ◆ 生成 .dcd 文件

标准驾驶员界面 (SDI) 和驾驶机器

- ◆ 标准驾驶员界面 (SDI) 为ADAMS/Car所用的一套结构体系，以完成利用输入来驱动你的虚拟样车的驾驶性能分析目的。这套体系可以进是ADAMS/Car 的一个特色，名为 Driving Machine，即驾驶机器。界面控制五个不同的信号：转向、油门、离合器、挡位和刹车。有了这些设置，使你很容易再现任何物理样机试验过程或从测试数据中复现实际的驾驶事件。
- ◆ 驾驶机器提供了与三种车辆控制方法的无缝接口：
 - **开环控制(Open-loop control)** – 开环控制可以使用常数或函数表达式来驱动车辆。没有反馈信号返回给控制系统。
 - **机器控制(Machine control)** – 机器控制为一种闭环控制，通过车辆的运动状态控制车辆。
 - **人工控制(Human control)** – 人工控制，类似于机器控制，为一种闭环控制，但是这种控制方式有学习的能力。

标准驾驶员界面 (SDI) 和驾驶机器

- ◆ 为了帮助你计算控制信号，驾驶机器传递车辆的状态，诸如，位置、速度和加速度给你的驾驶控制器。它同样提供一种方法以确定并传递系列命令信号、反馈信号以及五个控制信号对应的参数。
- ◆ 注意 ADAMS/Car 2005 版本引进控制和脚本仿真整车仿真的功能。在整车仿真对话框的外观没有什么改变，但内层的软件是新的。应用户的要求，MSC 开发了属于自己的算法代替了 IPG DriverLite，用于控制转向、调速和挡位变换，并采用一个 XML 格式的脚本控制整车的仿真，以替代原来的 TeimOrbit 格式的驾驶控制文件（driver control file 即 DCF文件）。

为什么使用 SDI?

■ 使用驾驶机器，你能做什么？

◆ 使用驾驶机器，你能够：

- 输入车辆的路径 $\{x, y\}$ 以及速度，使用闭环的驾驶机器以控制车辆的转向使其沿着给定的路径或按照给定的速度或二者兼而有之进行运动。
- 输入各种开环函数，如正弦扫频及脉冲信号，控制转向、油门或刹车等。
- 向模型输入记录的转向、油门、刹车、挡位和离合信号。
- 根据是否达到目标的横向加速度、纵向速度或移动的距离，停止仿真、切换控制器和改变输出步长。

为什么使用 SDI?

■ 使用驾驶机器能够得到什么？

◆ 使用驾驶机器，你有下述好处：

- 你可以无须再像以前那样设置闭环控制的正确的增益大小，从而可以节省时间。驾驶机器与 IPG 合作采用其所有的控制算法。
- 通过基于横向加速度、纵向速度或移动的位移达到一定水平作为仿真结束的目标条件，这样你可以节省仿真时间。
- 输入车辆的路径和所需要的速度，可以通过从存储在外部文件选择记录的数据，或基于几个样本输入数据点得到参数化生成车辆的路径和所需要的速度，你有了更大的灵活性。

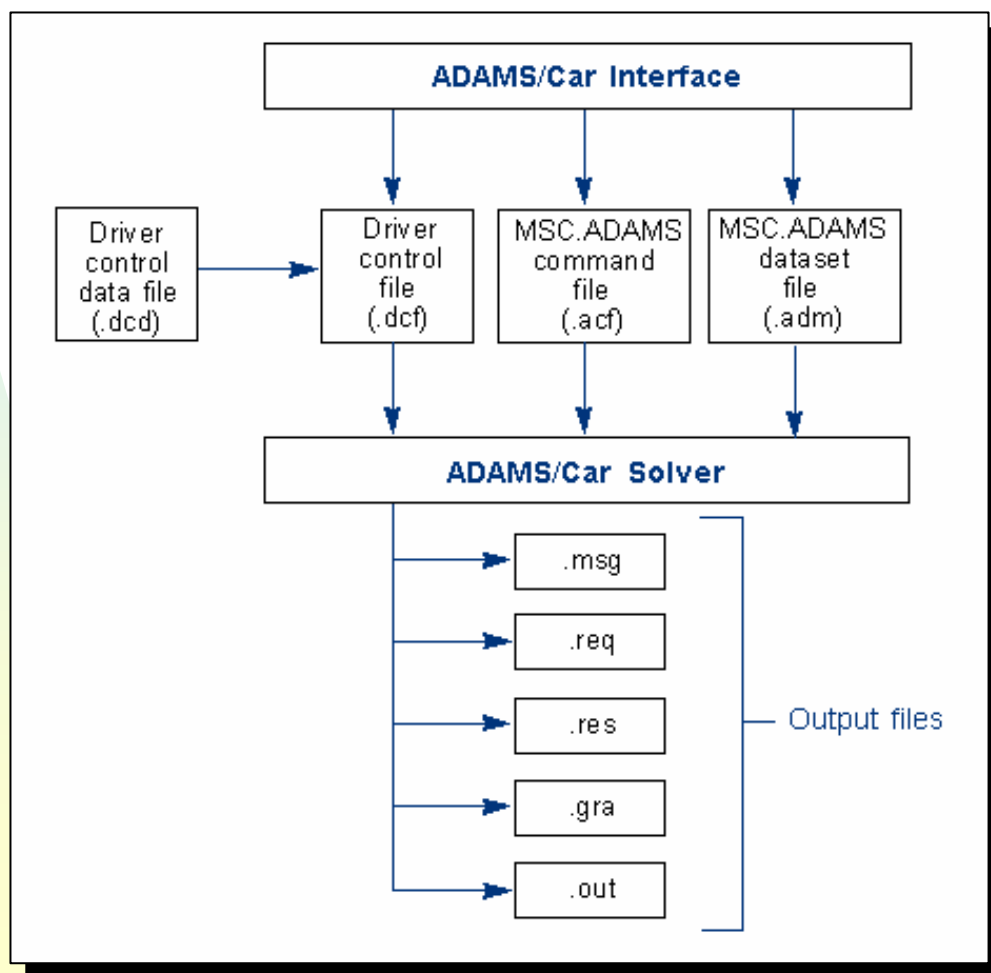
数据流程

■ 有关驾驶机器的数据流程

- ◆ 当你利用驾驶机器提交一个仿真时，ADAMS/Car 生成一个 ADAMS/Solver 的模型文件 (.adm)，一个 ADAMS/Solver 的命令文件 (.acf) 和一个驾驶控制文件 (.dcf)。驾驶控制文件 (.dcf) 为控制仿真事件的脚本。模型文件 (.adm) 参照驾驶控制文件 (.dcf)，而命令文件 (.acf) 调用一个包含在 ADAMS/Car 解算器中的 CONSUB 的子程序读取并处理 .dcf 文件中的内容以完成 .dcf 文件中脚本所描述的操控过程。ADAMS/Car 解算器然后提供标准的输出文件：.msg, .req, .res, .gra, and .out 等等。
- ◆ 驾驶控制文件 (.dcf) 描述你想执行的操控过程为一系列的最基本的操控过程。驾驶控制文件 (.dcf) 可以再参照一个或多个驾驶控制数据文件。驾驶控制数据文件包含或者机器驾驶（闭环控制）所需用的车辆路径和速度，或者车辆开环控制输入所需要的转向、油门、刹车、挡位和离合器的信号等。
- ◆ 下页汇总了驾驶机器所使用的数据流程。

数据流程

- 驾驶机器数据流程如下所示：



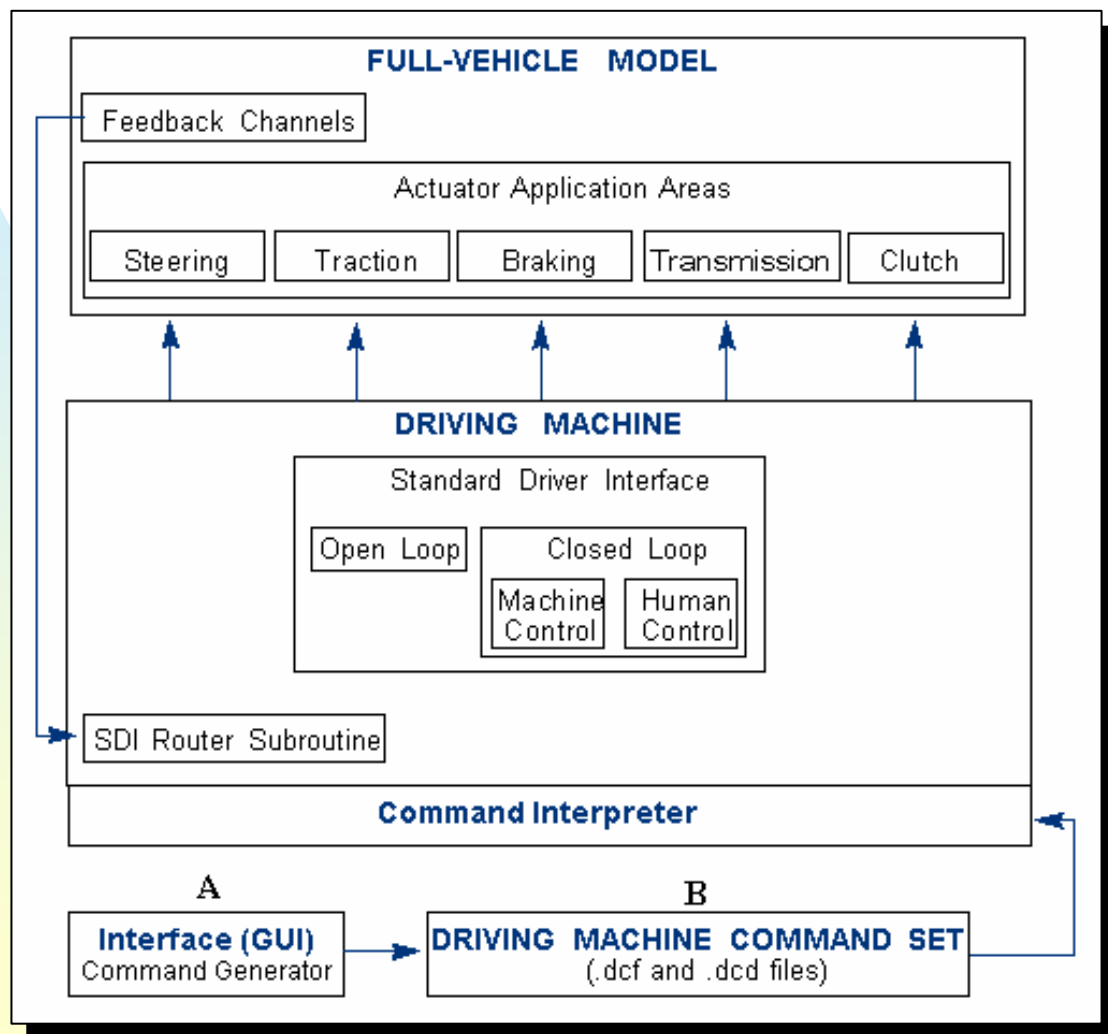
使用驾驶机器

■ 使用驾驶机器的步骤：

◆ 使用驾驶机器包含下列的步骤：

1. 将一个整车模型与 `.__MDI_SDI_TESTRIG` 试验台装配一起或打开一个已有的整车装配。
 2. 进行下面的一种分析：
 - 使用预先定义的分析
 - 生成你自己的 `.dcf` 文件以完成你所指定的一系列仿真过程
- ### ◆ 下页表示驾驶机器在虚拟样机模型中的工作机理。

使用驾驶机器



限制

- 驾驶机器有一些限制，如下所示：
 - ◆ 它只能在正转向输入使车辆向左转向时实现精确的转向。
 - ◆ 车速过低时控制无效。

生成 SDI 的输入

- 用于描述基于标准驾驶界面(SDI)事件的两种文件：驾驶控制文件 (.dcf) 和驾驶控制数据文件 (.dcd)。
 - ◆ 驾驶控制文件(.dcf) – 控制事件，包含一系列的最基本的控制，允许你用脚本描述仿真过程。
 - ◆ 驾驶控制数据文件(.dcd) – 包含用于驾驶控制文件(.dcf) 的输入数据。只有在驾驶控制文件(.dcf) 中需要参照时，才需要此文件。

驾驶员控制文件(Driver Control Files)

- ◆ 驾驶控制文件(.dcf) 描述在你想在虚拟试验或分析时驾驶机器如何驱动你的车辆。驾驶控制文件(.dcf) 指示驾驶机器以多快的速度驱动车辆、什么地方驱动车辆 (例如, 在 80m 半径的 skidpad), 并且停下来 (横向加速度=8m/s²时)。驾驶控制文件指定驾驶机器应该使用什么输入信号什么类型的控制, 如转向、油门等等。一个驾驶控制文件(.dcf)可以参照其它的文件, 主要是驾驶控制数据文件 (.dcd), 以得到必要的输入数据, 如 speed versus time等。



有关参照 .dcd 文件的信息, 请参见ADAMS/Car的在线帮助手册的 Analyze 部分。

- ◆ 驾驶控制文件将复杂的试验过程组织成一系列的小的简单的步骤称为最基本操控。一个驾驶控制文件主要包含一个静力学设置方法和一系列在 EXPERIMENT 块后面的最基本操控数据, 每一条对应一个最基本的操控方式。每一条最基本操控数据块依次指定驾驶机器如何控制转向、油门、刹车、挡位和离合器等。

驾驶员控制文件(Driver Control Files)

- ◆ 驾驶控制文件为 TeimOrbit 格式的 ASCII 文件，你可以在任何文本编辑器内进行修改。该文件允许你参照外部的 .dcd 文件以驱动车辆，如方向盘的转向位移、车辆的加速度或其它的输入。在驾驶控制文件中包含下面四个必须的数据块：
 - **MDI header** – 标识该文件为.dcf 文件并提供版本信息。
 - **Units** – 设置在 .dcf 文件中的单位，这些参数可以与模型中所使用的单位不同。
 - **Experiment** – 指定一些仿真的初始化条件，静力学求解设置选项和一系列的最基本操控命令，组成完整的试验或事件过程。
 - **Mini-maneuver(s)** – 指定车辆如何转向、多大油门、刹车参数、挡位和离合器等每个操控操作所要控制的参数，这些参数为车辆最基本的控制参数。最基本的操控的名字必须与 experiment 块下的名字相匹配。

驾驶员控制文件(.dcf)的例子

```
$-----MDI_HEADER
[MDI_HEADER]
FILE_TYPE      = 'dcf'
FILE_VERSION   = 2.0
FILE_FORMAT    = 'ASCII'
(COMMENTS)
{comment_string}
'DCF file for Open Loop Fish Hook Maneuver'

$-----UNITS
[UNITS]
LENGTH        = 'mm'
FORCE          = 'newton'
ANGLE          = 'deg'
MASS           = 'kg'
TIME           = 'sec'

$-----EXPERIMENT
[EXPERIMENT]
EXPERIMENT_NAME = 'Fish Hook Event'
INITIAL_SPEED   = 11111.11
INITIAL_GEAR    = 3
(MINI_MANEUVERS)
{mini_maneuver  abort_time  step_size}
'INITIAL_SET'   1.00        0.05
'FISH_HOOK'     4.00        0.02

$-----INITIAL_SET
[INITIAL_SET]
(STEERING)
ACTUATOR_TYPE   = 'ROTATION'
METHOD          = 'OPEN'
MODE            = 'ABSOLUTE'
CONTROL_TYPE    = 'CONSTANT'
CONTROL_VALUE   = 0
(THROTTLE)
METHOD          = 'MACHINE'
(BRAKING)
METHOD          = 'MACHINE'
(GEAR)
METHOD          = 'OPEN'
MODE            = 'ABSOLUTE'
CONTROL_TYPE    = 'CONSTANT'
CONTROL_VALUE   = 3
```

驾驶员控制文件(.dcf)的例子

```
(CLUTCH)
METHOD          = 'OPEN'
MODE            = 'ABSOLUTE'
CONTROL_TYPE    = 'CONSTANT'
CONTROL_VALUE   = 0
(MACHINE_CONTROL)
STEERING_CONTROL = 'STRAIGHT'
SPEED_CONTROL    = 'MAINTAIN'
(END_CONDITIONS)
{measure   test value  allowed_error  filter_time  delay_time  group}
'TIME'     '=='       1.0             0.0           0.0         0.0

$-----FISH_HOOK
[FISH_HOOK]
(STEERING)
ACTUATOR_TYPE   = 'ROTATION'
METHOD          = 'OPEN'
MODE            = 'ABSOLUTE'
CONTROL_TYPE    = 'FUNCTION'
CONTROL_EXPR    = 'STEP(TIME,1.0,0.0,1.20,10.000000)+STEP(TIME,1.70,0.0,2.50,90.000000)'
(THROTTLE)
METHOD          = 'OPEN'
MODE            = 'ABSOLUTE'
CONTROL_TYPE    = 'FUNCTION'
CONTROL_EXPR    = '0.0'
(BRAKING)
METHOD          = 'OPEN'
MODE            = 'ABSOLUTE'
CONTROL_TYPE    = 'FUNCTION'
CONTROL_EXPR    = '0.0'
(GEAR)
METHOD          = 'OPEN'
MODE            = 'ABSOLUTE'
CONTROL_TYPE    = 'CONSTANT'
CONTROL_VALUE   = 3
(CLUTCH)
METHOD          = 'OPEN'
MODE            = 'ABSOLUTE'
CONTROL_TYPE    = 'CONSTANT'
CONTROL_VALUE   = 1
(END_CONDITIONS)
{measure   test value  allowed_error  filter_time  delay_time  group}
'TIME'     '=='       4.0             0.0           0.0         0.0
```

生成 .dcf 和 .dcd 文件

- 生成一个 .dcf 文件最简单的方法就是将一个共享数据库下的 .dcf 文件拷贝下来，再作数据块必要的修改。
下面讲述对数据块可做的修改：
 - ◆ **MDI header block** – 这部分不需要做过多的修改：只需要保证你所使用的 .dcf 文件与你打算使用的 ADAMS/Car 为同一版本。
 - ◆ **Units block** – 单位应该为在你的 .dcf 文件中所使用的数据的单位，可以用于你的装配模型中所使用的单位不同。
 - ◆ **Experiment block** – 如下图所示：

```
$-----EXPERIMENT
[EXPERIMENT]
EXPERIMENT_NAME    = 'Fish Hook Event'
INITIAL_SPEED      = 11111.11
INITIAL_GEAR       = 3
(MINI_MANEUVERS)
{mini_maneuver      abort_time    step_size}
'INITIAL_SET'       1.00          0.05
'FISH_HOOK'         4.00          0.02
```

驾驶员控制文件(.dcf)的结构

- ◆ EXPERIMENT 数据块 – 指定静力学求解设置，初始速度，初始挡位以及一系列试验的最基本操控方式。驾驶机器按照文件中列出顺序执行操控过程，直到列表操控过程结束或某一个最基本操控过程被终止。

在 EXPERIMENT 数据块，你必须提供 INITIAL_SPEED 的参数，可选的，INITIAL_GEAR 的参数 (缺省值为 3) 以及 INITIAL_CLUTCH 的参数 (缺省值为 0，保持啮合状态)。可选地，你能够指定静力学求解的设置，这样可以除去开始仿真阶段的瞬态影响，也可以减少你正常需要的最小操控步骤，比如，你要做一次转圈操控分析。

你可以在 STATIC_SETUP 下选择下面的任何值。如果你忽略 STATIC_SETUP 的值，ADAMS/Car 使用缺省值 NORMAL。

- NORMAL
- NONE
- SETTLE
- STRAIGHT
- SKID_PAD



有关 EXPERIMENT 数据块方面更多信息，参见 ADAMS/Car 在线帮助手册 Analyze 部分。

驾驶员控制文件(.dcf)的结构

在 EXPERIMENT 数据块的后面必须带有 MINI_MANEUVER 子数据块。在每一个基本操控中，MINI_MANEUVER子数据块必须指定一个基本操控的名字、结束时间和输出步长。可选地，你能够定义每个基本操控的结束条件，最大的积分步长 HMAX (HMAX必须小于输出步长)。如果结束时间先于满足结束条件，驾驶机器将终止整个仿真试验过程，而不仅仅只是该基本操控步骤。你也可以指定结束条件子数据块，该结束条件子数据块可以应用于试验过程中所有的基本操控步骤。

- ◆ 下图所示为一个做好的完整的 EXPERIMENT 数据块的例子。

```
[EXPERIMENT]
EXPERIMENT_NAME    = 'Braking in a Turn'
INITIAL_SPEED      = 17152.55
INITIAL_GEAR       = 3

(MINI_MANEUVERS)
{mini_maneuver      abort_time    step_size    hmax}
'INITIAL_SET'       10.00         0.05         0.02
'BRAKE_TURN'        15.00         0.02
```

驾驶员控制文件(.dcf)的结构

- ◆ Mini-maneuver block(s) – 一个驾驶控制文件可以包含的基本操控步骤的数量并没有限制。每一个基本操控块必须以在 EXPERIMENT 数据块下的 MINI_MANEUVERS 子数据块内所指定的列表作为标题。每个基本操控块必须包含子数据块，其标题为 (STEERING)、(THROTTLE)、(BRAKE)、(GEAR) 和 (CLUTCH)。依据你指定的控制方法，你可能需要包含或者 MACHINE_CONTROL或者 HUMAN_CONTROL数据块。一个基本操控块同样可以包含一个 END_CONDITIONS数据块，其中指定一个结束条件，可以作为触发条件结束该基本操控步骤，并转到下一个操控步骤。通常来讲，这样的数据块对所有的基本操控步骤都是需要的，而不只是最后一个基本操控步骤，这样可以保证所有的基本操控步骤在结束时间到了以前都能够被执行。
- ◆ 下页所示为一个基本操控块的例子，基本操控块的名字为 BRAKE_TURN：

驾驶员控制文件(.dcf)的结构

```
$-----BRAKE_TURN
[BRAKE_TURN]
(STEERING)
ACTUATOR_TYPE   = 'ROTATION'
METHOD          = 'OPEN'
MODE            = 'RELATIVE'
CONTROL_TYPE    = 'CONSTANT'
CONTROL_VALUE   = 0

(THROTTLE)
METHOD          = 'MACHINE'

(BRAKING)
METHOD          = 'MACHINE'

(GEAR)
METHOD          = 'OPEN'
MODE            = 'ABSOLUTE'
CONTROL_TYPE    = 'CONSTANT'
CONTROL_VALUE   = 3

(CLUTCH)
METHOD          = 'OPEN'
MODE            = 'ABSOLUTE'
CONTROL_TYPE    = 'CONSTANT'
CONTROL_VALUE   = 0

(MACHINE_CONTROL)
SPEED_CONTROL   = 'VEL_POLYNOMIAL'
VELOCITY        = 17152.55
ACCELERATION    = -1961.40
JERK            = 0.0
START_TIME      = 1.0

(END_CONDITIONS)
{measure      test value   allowed_error  filter_time  delay_time  group}
'VELOCITY'    '<<'        2500.00    0.0          0.0         0.0
```

指定结束条件

- 对每一个基本操控步骤，你可以在一个子数据块内指定结束仿真条件。END_CONDITIONS 子数据块看起来应该如下所示：

```
(END_CONDITIONS)
{measure      test      value      allowed_error  filter_time  delay_time  group}
'LAT_ACCEL'   '=='      5884.20   0.0           2.00         0.0
```

其中：measure 为监测量，比如：

- ◆ VELOCITY – 车辆的纵向速度
- ◆ LAT_VELOCITY (VY) – 车辆的横向速度
- ◆ VERT_VELOCITY (VZ) – 车辆的垂向速度
- ◆ LON_ACCEL - 车辆的纵向加速度
- ◆ LAT_ACCEL - 车辆的横加向速度
- ◆ VERT_ACCEL (ACCZ) - 车辆的垂向加速度

指定结束条件

- ◆ DISTANCE – 在基本操控期间车辆所驶过的总里程
- ◆ TIME – 仿真时间
- ◆ YAW_ANGLE – 相对于车辆垂直轴转动的角位移
- ◆ YAW_RATE -相对于车辆垂直轴转动的角速度
- ◆ YAW_ACCEL -相对于车辆垂直轴转动的角加速度
- ◆ PITCH_ANGLE -相对于车辆横向坐标轴转动的角位移
- ◆ ROLL_ANGLE -相对于车辆纵向坐标轴转动的角位移
- ◆ SIDE_SLIP_ANGLE - Angular offset between the projected vehicle's longitudinal axis and the vehicle's velocity vector(侧偏角)
- ◆ LON_DIS (DX) – 纵向位移
- ◆ LAT_DIS (DY) – 横向位移

指定结束条件

- ◆ VERT_DIS (DZ) – 垂向位移
- ◆ ROLL_RATE (WX) - 相对于车辆纵向坐标轴转动的角速度
- ◆ PITCH_RATE (WY) -相对于车辆横向坐标轴转动的角速度
- ◆ ENGINE_SPEED – 发动机曲轴的角速度，转/分钟 (rpm)
- ◆ RACK_TRAVEL – 在转向拉杆连接点处的位移
- ◆ STEERING_ANG – 在方向盘上的角位移

■ 当前测试的可选项有：

- ◆ test – 可以：
 - ‘==’ 等于 (+/- 0.2%。例如，如果基本操控步骤中的测试为 VELOCITY，而值为 10 m/s的话，驾驶机器将在车辆的纵向速度测试值大于等于9.98 m/s 及小于等于 10.02 m/s 时结束该基本操控步骤。)
 - ‘>>’ 大于
 - ‘<<’ 小于

指定结束条件

- ◆ value – 阈值，确定结束条件是否满足。除了 ENGINE_SPEED 之外，你必须按照模型中的单位指定, you must specify the value in modeling units as defined in the UNITS block of the driver control file.
- ◆ allowed_error – 测试数据和满足测试的阈值之间允许误差。允许误差只能为正值，measure and value that still satisfies the test. allowed_error must be positive and be specified in modeling units as defined in the UNITS block of the driver control file (except ENGINE_SPEED). 例如：

LAT_ACCEL '==' 5 0.1 0.0 0.0

意味着结束条件将满足，当：

$5 - 0.1 < \text{Lateral acceleration} < 5 + 0.1$

或结束条件为：

LAT_ACCEL '>>' 5 0.1 0.0 0.0

意味着结束条件将满足，当 lateral acceleration $> 5 - 0.1$ 。

或结束条件为：

LAT_ACCEL '<<' 5 0.1 0.0 0.0

意味着结束条件将满足，当 lateral acceleration $< 5 + 0.1$ 。

指定结束条件

- ◆ filter_time – 过滤时间窗，测试必须在过滤时间窗内连续地满足结束条件。过滤时间窗只能为正。
- ◆ delay_time – 时间延迟，一旦满足结束条件，将按照延迟时间延迟基本操控的结束时间。
- ◆ group – 组，你可以指定若干个结束条件为一个组，并赋其一个名字。具有同一组名的所有的结束条件必须同时满足才结束基本操控步骤。例如，你可以指定两个结束条件：
 - Longitudinal velocity 等于 20 m/s
 - Lateral acceleration 大于 5 m/s/s

你可以将结束条件放在一个名为 'mygroup' 的组下。要结束基本操控条件，longitudinal velocity 必须为 20 m/s，同时lateral acceleration 必须大于 5 m/s/s。如下所示编辑结束条件：

VELOCITY '==' 20 0.1 0.0 0.0 'mygroup'

LAT_ACCEL '>>' 0 0.1 0.0 0.0 'mygroup'

指定结束条件

要定义不同的子数据块，你必须指定其属性。下表列出你必须给每个子数据块指定的属性：

These sub-blocks:	Require these attributes:
Steering	<ul style="list-style-type: none">■ ACTUATOR_TYPE■ METHOD■ MODE*
Throttle, brake, gear, and clutch	<ul style="list-style-type: none">■ METHOD■ MODE*

* 只在开环控制方法需要。

- ◆ 下页描述你可以指定的属性。

可扩展的结束和退出条件

- ◆ 一个条件传感器为用户定义的要素，包含一个数据元素数组和一个字符串。它参考一个已经存在的变量类要素(数据要素变量或Solver计算的测试)，将其捆绑在一起到一个标签和单位字符串数组下。该数组也捆绑一个测试请求 (为绘图方便) 和一个单位转换因子。
- ◆ 从本质上讲，一个条件传感器代表一种可测试得到的 Solver量(变量类型对象)和一个字符串标签标识之间的关系，此字符串标签标识可以被用在驾驶控制文件 (.dcf) 中，以定义一个 ADAMS/Car 整车分析的结束条件。

■ 在 ADAMS/Car 中条件传感器的使用

- ◆ 在每次车辆分析之前，ADAMS/Car 浏览装配，以找到条件传感器并以得到的数据更新数据要素 end_conditions_array。在仿真开始阶段，标准驾驶界面 (SDI) 使用在驾驶控制文件 (.dcf) 中特定的结束条件测试字符串，以标识在模型文件中相关的变量类型对象，在仿真过程中计算结束条件传感器应该与目标值对比的量。

指定属性

■ 指定一个激励类型

- ◆ 在定义一个 STEERING 的子数据块时，你必须指定要实现转向所需要的激励类型。使用激励类型以指定驾驶机器如何使车辆转向，是作用在方向盘上，还是作用在转向拉杆上；是用力驱动还是用运动驱动，具体的内容将在后面详细描述。激励包括：转动 (加在方向盘上)、移动 (加在转向拉杆上)、力 (加在转向拉杆上)或力矩 (加在方向盘上)。

■ 指定一个控制方法

- ◆ 当定义任何的子数据块时，你必须指定想用的控制方法。你可以从下面三种方法中进行选择：
 - OPEN_CONTROL：开环控制
 - MACHINE_CONTROL：机器控制
 - HUMAN_CONTROL：人工控制

指定属性

■ 开环控制

- ◆ 当你指定 METHOD = 'OPEN' 为转向或任何其它信号时，驾驶机器输出信号为时间的函数，并且你必须使用CONTROL_TYPE 参数指定函数的类型。

- **Arguments:**

CONTROL_TYPE = 'CONSTANT' || 'STEP' || 'RAMP' || 'IMPULSE' ||
'SINE' || 'SWEPT_SINE' || 'DATA_DRIVEN' || 'FUNCTION'

■ 机器控制

- ◆ 在 .dcf 文件中，MACHINE_CONTROL 子数据块指定车辆要走的路径、速度和其它机器控制基本操控所需要的参数。

指定属性

- ◆ 我们推荐，如果你使用机器控制油门，你同样要使用机器控制刹车。同样地，如果使用机器控制挡位，你也应该使用机器控制离合器。当你选择 MACHINE_CONTROL 控制挡位和离合器，你必须再提供发动机的最高最低转速。机器控制将自动换档以使发动机的输出转速不超过最高转速或向下换档以不低于最低转速。
- ◆ 当使用机器控制时，你必须保证你的 .dcf 文件有一个 MACHINE_CONTROL 的子数据块，在 MACHINE_CONTROL 的子数据块中，指定速度如何控制和转向如何控制。
- ◆ 如果 SPEED_CONTROL = LAT_ACCEL，你必须设置 STEERING_CONTROL 为 SKIDPAD。
- ◆ 参数：
 - STEERING_CONTROL = 'FILE' || 'STRAIGHT' || 'SKIDPAD'
 - SPEED_CONTROL = 'FILE' || 'LAT_ACCEL' || 'MAINTAIN' || 'VEL_POLYNOMIAL' || 'LON_ACCEL'

指定属性

■ 人工控制

- ◆ 在 .dcf 文件的 HUMAN_CONTROL 子数据块指定 ADAMS/Driver 的输入文件和参数。ADAMS/Driver 与真人驾驶员的行为相类似并具有学习的能力，能够适应不同车辆的特点。在使用 ADAMS/Driver 之前，需要系统的了解 ADAMS/Driver 的功能，我们推荐你读一下 ADAMS/Driver 在线帮助手册 *The Concept of ADAMS/Driver* 部分。
- ◆ 要使用 HUMAN_CONTROL，你必须有一个 ADAMS/Driver 的单独的 license 才可以。

指定属性

■ 指定属性

- ◆ 当你定义任何的子数据块并使用 METHOD='OPEN'，你必须定义 MODE 为相对的(relative) 或绝对的 (absolute)。对所有的开环操控而言，前一步基本操控的结果将作为下一步基本操控的起点。这与你选择相对的(relative) 还是绝对的 (absolute) 没有关系。例如：你的车辆在 skid pad 上以 30 mph速度行驶，方向盘的转向角为 20° ，当第一步基本操控结束时，下一步基本操控的方向盘的转向角为 20° 不变。
- ◆ 相对的(relative) 或绝对的 (absolute) 方法允许你定义在下一步基本操控结束时方向盘的转向角为多少。

驾驶员控制数据文件 (Driver Control Data Files)

- 驾驶员控制数据文件包含驾驶机器所使用的数据。为指示驾驶机器引用 .dcd 文件中的数据，你必须在驾驶控制文件 (.dcf) 中引用这些文件。在驾驶控制文件 (.dcf) 中如何引用这些 .dcd 文件的例子如下所示：

```
(STEERING)
METHOD          = 'OPEN'
CONTROL_TYPE    = 'DATA_DRIVEN'
FILE_NAME       = 'my_data.dcd'
```


驾驶员控制数据文件 (Driver Control Data Files)

- 驾驶员控制数据文件有两种类型的数据，如下所示：
 - ◆ **开环数据 (Open-loop data)** – 作为控制车辆行驶的输入数据，不考虑车辆行驶的速度，也不考虑车辆行驶的方向。相关的数据包括：方向盘转向角、油门、刹车、挡位和离合器信号。开环数据的一个例子中包含方向盘转向角 vs. 时间和油门位置 vs. 时间的数据。
 - ◆ **闭环数据 (Closed-loop data)** – 精确指定车辆将行驶的方向和速度的数据。闭环数据的例子如提供车辆的 x 和 y 的位置坐标 vs. 时间。你有几种方式指定车辆的闭环数据。例如，曲率和速度 vs. 前进的距离或横向加速度和纵向加速度 vs. 时间。你使用 .dcd 文件中的 SPEED_CONTROL 和 STEERING_CONTROL 属性指定数据的类型。

驾驶控制数据文件(Driver Control Data Files)的构造体系

- 驾驶控制数据 (.dcd) 文件使用 TeimOrbit 格式，类似于 ADAMS/Car 其它的属性文件。驾驶控制数据 (.dcd) 文件必须包含下面这些数据块：
 - ◆ **MDI_HEADER** 数据块 - 标识此文件为 .dcd (而不是一个 .dcf) 文件并提供版本信息。
 - ◆ **UNITS** 数据块 - 标识包含在 .dcd 文件中的数据的单位。
- 驾驶控制数据 (.dcd) 文件同时必须包含下列两类数据块中的一个：
 - ◆ **OPEN-LOOP** 数据块- 指定车辆的转向角、油门、刹车、挡位和离合器等输入信号。
 - ◆ **CLOSED-LOOP** 数据块-指定车辆的行驶路径或速度或二者都有。

驾驶控制数据文件(Driver Control Data Files)的构造体系

- 驾驶控制数据文件也可以既包含 OPEN-LOOP 数据块又包含 CLOSED-LOOP 数据块
- 下页的表汇总了 .dcd 文件中可以包含的CLOSED-LOOP 数据。列代表驾驶参数数组中速度控制的选项。行代表驾驶参数数组中转向控制的选项。交叉项给出包含在 .dcd 文件中的数据，因此，输入的数据为 Driver-Lite 所需要的数据以产生 {x, y, lon_vel}。

驾驶员控制数据文件(Driver Control Data Files)的构造体系

SPEED_CONTROL STEERING_CONTROL	none	lon_vel (p1=0)	lon_acc (p1=1)	lat_acc (p1=2)	path (p1=3)
none	NOT VALID	{{(distance or time), lon_vel}}	{{(distance or time), lon_acc}}	NOT VALID	NOT VALID
curvature (p1 = 0)	{distance, curvature}	{{(distance or time), curvature, lon_vel}}	{{(distance or time), curvature, lon_acc}}	{{(distance or time), curvature, lat_acc}}	NOT VALID
path (p1 = 1)	{x, y}	{x, y, lon_vel}	{x, y, lon_acc}	{x, y, lat_acc}	{x, y, time}
lat_acc (p1 = 2)	NOT VALID	{distance or time, lat_acc, lon_vel}	{distance or time, lat_acc, lon_acc}	NOT VALID	NOT VALID

生成 .dcd 文件

- ◆ 你可以使用MSC.Software 提供的 .dcd 文件的样板或生成你自己的 .dcd 文件。

■ 要生成 .dcd 文件：

1. 使用下面三种方式中的一种生成 .dcd 文件：
 - 运行一次实际的或虚拟的试验并记录所得到的五个控制信号数据。
 - 在文本编辑器内，修改我们提供给你的 .dcd 文件样板。
 - 使用文本编辑器生成一个 .dcd 文件，并按照 ADAMS/Car 在线帮助手册中 Analyze 部分的指导和格式进行编辑修改。
2. 存储此文件并在一个 .dcf 文件中引用该文件。

练习 8 编辑 .DCF 和 .DCD 文件

■ 问题阐述

- ◆ 在本练习中，你将学习如何在缺省的 ADAMS/Car 的共享数据库中已有的文件基础上编辑一个 .dcf 和 .dcd 文件。



本练习大约需要一个小时完成。

练习 8 编辑 .DCF 和 .DCD 文件

■ .dcf 文件的例子

```
$-----MDI_HEADER
[MDI_HEADER]
FILE_TYPE           = 'dcf'
FILE_VERSION        = 2.0
FILE_FORMAT         = 'ASCII'
(COMMENTS)
{comment_string}
'DCF file for Closed Loop ISO-Lane Change'
$-----UNITS
[UNITS]
LENGTH              = 'meters'
FORCE                = 'newton'
ANGLE                = 'deg'
MASS                 = 'kg'
TIME                 = 'sec'
$-----EXPERIMENT
[EXPERIMENT]
EXPERIMENT_NAME      = 'ISO-Lane Change'
STATIC_SETUP         = 'STRAIGHT'
INITIAL_SPEED        = 16.667
INITIAL_GEAR         = 3
(MINI_MANEUVERS)
{mini_maneuver      abort_time  step_size}
'LANE_CHANGE'       12.0        0.05
$-----LANE_CHANGE
[LANE_CHANGE]
(STEERING)
ACTUATOR_TYPE        = 'ROTATION'
METHOD                = 'MACHINE'
```

练习 8 编辑 .DCF 和 .DCD 文件

```
(THROTTLE)
METHOD          = 'MACHINE'
(BRAKING)
METHOD          = 'MACHINE'
(GEAR)
METHOD          = 'OPEN'
MODE            = 'ABSOLUTE'
CONTROL_TYPE    = 'CONSTANT'
CONTROL_VALUE   = 3
(CLUTCH)
METHOD          = 'OPEN'
MODE            = 'ABSOLUTE'
CONTROL_TYPE    = 'CONSTANT'
CONTROL_VALUE   = 0
(MACHINE_CONTROL)
STEERING_CONTROL = 'FILE'
DCD_FILE         = '<acar_shared>/driver_data.tbl/iso_lane_change.dcd'
SPEED_CONTROL    = 'MAINTAIN'
MIN_ENGINE_SPEED = 750.0
MAX_ENGINE_SPEED = 6500.0
(END_CONDITIONS)
{measure      test value allowed_error filter_time delay_time group}
'DISTANCE'   '=='      250000.00    500.0    0.0    0.0
```


练习 8 编辑 .DCF 和 .DCD 文件

- ◆ 右图为进行 ISO变线分析所用的 .dcd 文件：

```
[MDI_HEADER]
FILE_NAME           = iso_lane_change.dcd
FILE_TYPE           = 'dcd'
FILE_VERSION        = 1.0
FILE_FORMAT         = 'ASCII'
(COMMENTS)
{comment_string}
'Example DCD file of ISO-Lane Change Path'

$-----UNITS

[UNITS]
LENGTH   = 'meters'
FORCE    = 'newton'
ANGLE    = 'radians'
MASS     = 'kg'
TIME     = 'sec'

$-----CLOSED_LOOP

[CLOSED_LOOP]
STEERING_CONTROL = 'path'
SPEED_CONTROL    = 'none'

(DATA)
{X      Y}
0.0     0.000
45.0    0.000
52.5    0.000
60.0    0.000
90.0    3.586
102.0   3.586
115.0   3.586
140.0   0.172
147.0   0.172
155.0   0.172
162.0   0.172
170.0   0.172
200.0   0.172
300.0   0.172
400.0   0.172
500.0   0.172
```

练习 8 编辑 .DCF 和 .DCD 文件

■ 借用缺省的 .dcf 文件

- ◆ 在这一节中，你将打开一个装配模型，使用缺省的 .dcf 文件完成一个整车的装配分析，并观察仿真结果。

■ 要打开一个装配模型：

1. 从菜单 **File** 下，指向 **Open**，然后选择 **Assembly**。
2. 搜索共享数据库并打开装配 **MDI_Demo_Vehicle.asy**。

◆ 要完成整车分析：

1. 从菜单 **Simulate** 下，指向 **Full-Vehicle Analysis**，然后选择 **DCF Driven**。
2. 在 **Output Prefix** 文本栏内，输入 **default**。
3. 在 **Driver Control Files** 文本栏内点击鼠标右键，并选择浏览 **<acar_shared>\driver_controls.tbl** 下，双击文件 **iso_lane_change.dcf**。
4. 选择 **OK**。

练习 8 编辑 .DCF 和 .DCD 文件

■ 要观察仿真结果：

1. 启动 ADAMS/PostProcessor。
2. 从菜单 **Simulation** 列表中选择 **default_iso_lane_change**。
3. 从 **Filter** 列表选择 **user_defined**。
4. 从 **Request** 列表选择 **chassis_displacements**。
5. 从 **Component** 列表选择 **lateral**。
6. 选择 **Add Curves**。

曲线图中显示底盘的横向位移曲线，与你在驾驶控制数据文件中的相同。显示的曲线表示车辆从一条车道变到另一条车道再回到原来的车道上的过程。

7. 关闭 ADAMS/PostProcessor。

练习 8 编辑 .DCF 和 .DCD 文件

■ 要拷贝 .dcf 和 .dcd 文件到你自己的数据库下：

1. 从菜单 **Tools** 下指向 **Database Management**，然后选择 **Database Info** 检查一下共享数据库的路径。
2. 从共享数据库下，拷贝文件 **iso_lane_change.dcf** 文件到你的 **acar_training** 数据库下。
3. 将其改名为 **iso_lane_change_wider.dcf**。
4. 从共享数据库下，拷贝文件 **iso_lane_change.dcd**到你的 **acar_training** 数据库下。
5. 将其改名为 **iso_lane_change_wider.dcd**。

练习 8 编辑 .DCF 和 .DCD 文件

■ 编辑 .dcd 和 .dcf 文件

- ◆ 修改 .dcd 文件，增大车道位移的变化量，然后改变你的.dcd 和 .dcf 文件中的名字以使与其新的名字相对应。

*注意：*在编辑 .dcd 文件之前，编辑 iso_lane_change_wider.dcf 文件，改变在 LANE_CHANGE 基本操控块下面文件的名字。

■ 要编辑文件 iso_lane_change_wider.dcd：

- ◆ 改变在 Y 列的三个输入数据，从 3.5 改为 10，如下图所示：

```
$-----CLOSED_LOOP
[CLOSED_LOOP]
STEERING_CONTROL = 'path'
SPEED_CONTROL    = 'none'

(DATA)
{ X      Y }
  0.0    0.000
 45.0    0.000
 52.5    0.000
 60.0    0.000
 90.0    10
102.0    10
115.0    10
140.0    0
...
```

*注意：*在 [MDI_HEADER]下的
FILE_NAME不需要与实际
的文件名相同。

练习 8 编辑 .DCF 和 .DCD 文件

■ 运行一次仿真并观察结果

- ◆ 使用你的 acar_training 数据库下的修改的 .dcf 文件运行一次整车的仿真。

■ 要运行一次整车的仿真：

1. 从菜单 **Simulate** 下，指向 **Full-Vehicle Analysis**，并选择 **DCF Driven**。
2. 在 **Output Prefix** 文本栏内输入 **wider_run**。
3. 在 **Driver Control File** 文本栏内点击鼠标右键并选择 **Search**，然后再选择 **<acar_training>\driver_controls.tbl**。
4. 在文件 **iso_lane_change_wider.dcf** 上双击鼠标左键。
5. 选择 **OK**。

练习 8 编辑 .DCF 和 .DCD 文件

■ 要观察仿真结果：

1. 启动 ADAMS/PostProcessor。
你应该可以见到第一次 lane-change 仿真的结果曲线还在。
2. 从 **Simulation** 列表下选择 **wider_run**。
3. 从 **Filter** 列表下选择 **user_defined**。
4. 从 **Request** 列表下选择 **chassis_displacements**。
5. 从 **Component** 列表下选择 **lateral**。
6. 选择 **Add Curves**。

曲线图表示车辆比第一次仿真偏移的距离更大一些。从结果上看，并非精确地满足要求的偏移距离 10 m (10000 mm)，因为闭环控制所执行的操控过程是要提供要求的输出和仿真条件最佳的可能响应。因此，如果我们给驾驶机器更多的时间和更大的距离来完成这一操控过程，结果将会更接近于所要求的 10 m 的偏移量。

练习 8 编辑 .DCF 和 .DCD 文件

例如，如果你按照下图修改你的 .dcd 文件，你将看到横向位移file as shown next, you would see that the lateral displacement would be closer to the desired value of 10 m:

(DATA)	
{ X	Y }
0.0	0.000
45.0	0.000
52.5	0.000
60.0	2
90.0	4
102.0	6
115.0	8
140.0	10
147.0	10
155.0	8
162.0	6
170.0	4
200.0	2
300.0	0.172
400.0	0.172
500.0	0.172

- ◆ 有关驾驶机器更多的信息，参见 ADAMS/Car的在线帮助手册 Analyze 部分。如果你有时间的话，运行并修改 ADAMS/Car 共享数据库下其它的 .dcf 文件。

第 9 章 参数化

- 本章主要讲述 ADAMS/Car 是如何使用硬点和构造坐标系通过位置、方向实现模型参数化的。

参数化

- 本章内容：
 - ◆ 在 ADAMS/Car 中的参数化
 - ◆ 生成硬点
 - ◆ 生成构造坐标系
 - ◆ 位置参数化
 - ◆ 方向参数化

在 ADAMS/Car 中的参数化

■ 为什么要参数化？

- ◆ 参数化一个模板可以使你建立模型中的关联关系，这样，当你改变模型中的一个对象，ADAMS/Car自动地对模型中的其它相关的对象进行更新。由此，你能够建立一个完整的参数化的车辆模型，该模型只取决于少量的关键参数，如关键硬点和变量，这样的话，在修改设计时，可以节省很多时间和工作量。

■ 什么东西你可以参数化？

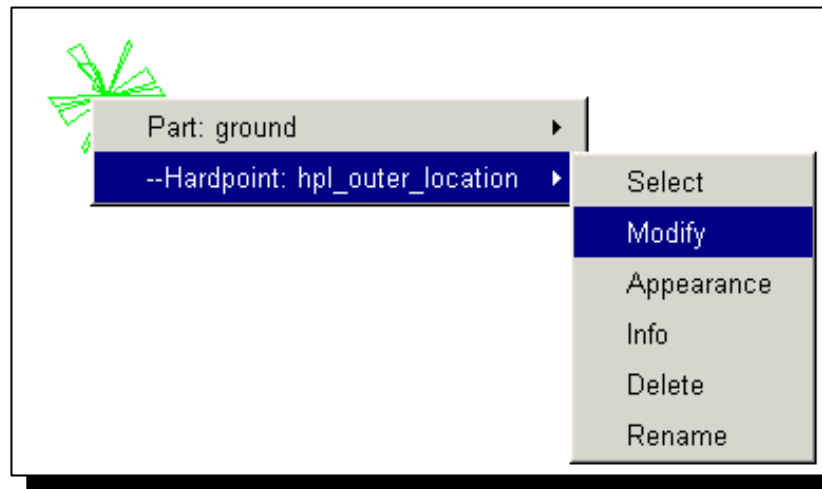
- ◆ 位置和方向表达式
- ◆ 几何外形
- ◆ 组的激活
- ◆ 函数
- ◆

生成硬点

- 硬点定义模型中所有的关键点。硬点为参数化建模最常用的元素，用于更高层次的对象的位置的参数化，如构造坐标系、部件和联结点等等。硬点跟 ADAMS/View 中的设计点是一样的，但硬点，按照缺省方式是总是建在大地上的。
- 你可以在模板建模方式下生成硬点。要生成硬点，可以从 Build → Hardpoint → New。在对话框中，指定硬点的名字，左、右还是单个硬点以及其位置。当你生成的硬点是左侧或右侧的应点时，ADAMS/Car会相应的生成另外一侧沿着车辆纵轴对称的硬点。

生成硬点

- 你可以通过在硬点上点击鼠标右键的方式修改硬点。



- 同样你可以修改硬点，通过 Build → Hardpoint，你可以选择：
 - ◆ **Table** – 显示模型中所有的硬点，你可以输入每个硬点的新位置。
 - ◆ **Modify** – 你一次只能修改一个硬点的位置。

生成构造坐标系

- 构造坐标系为一类建模元素，用于一个高层次的对象的参数化时，不单单要指定位置还要指定方向。构造坐标系与 ADAMS/View中的坐标系标记点是一样的。
- 要生成构造坐标系，可以从 Build → Construction Frame → New。在对话框内指定其名字，左右或单个性，要定义其位置和方向，可以选择下面的选项：

生成构造坐标系

■ 位置参数化

- ◆ 相对一个局部坐标系的位置
- ◆ 坐标系的中间位置
- ◆ 在两个坐标系的中间连线上
- ◆ 沿着一个坐标系的某一轴上
- ◆ 通过信息交换器输入的位置
- ◆ 在弹性体的节点上

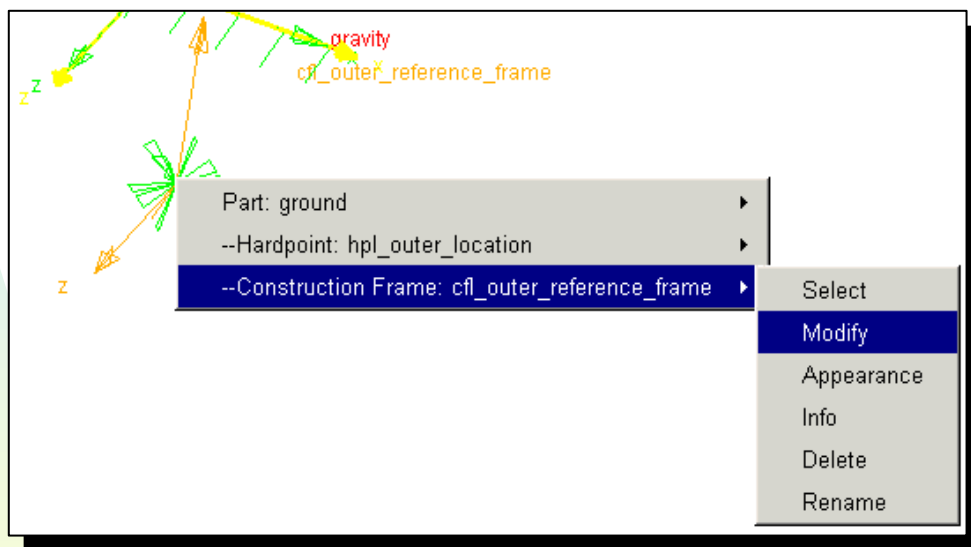
生成构造坐标系

■ 方向参数化

- ◆ 用户输入数值 (欧拉角)
- ◆ 相对一个局部坐标系的方向角
- ◆ 平行于某个轴
- ◆ 在一个平面内
- ◆ 方向指向由 $zpoint - xpoint$ 确定
- ◆ 方向沿着一条直线
- ◆ 方向轴指向点
- ◆ 方向输入信息交换器
- ◆ Toe/camber
- ◆ 相对轴的方向

生成构造坐标系

- 你可以通过在构造坐标系上点击鼠标右键的方式修改构造坐标系。

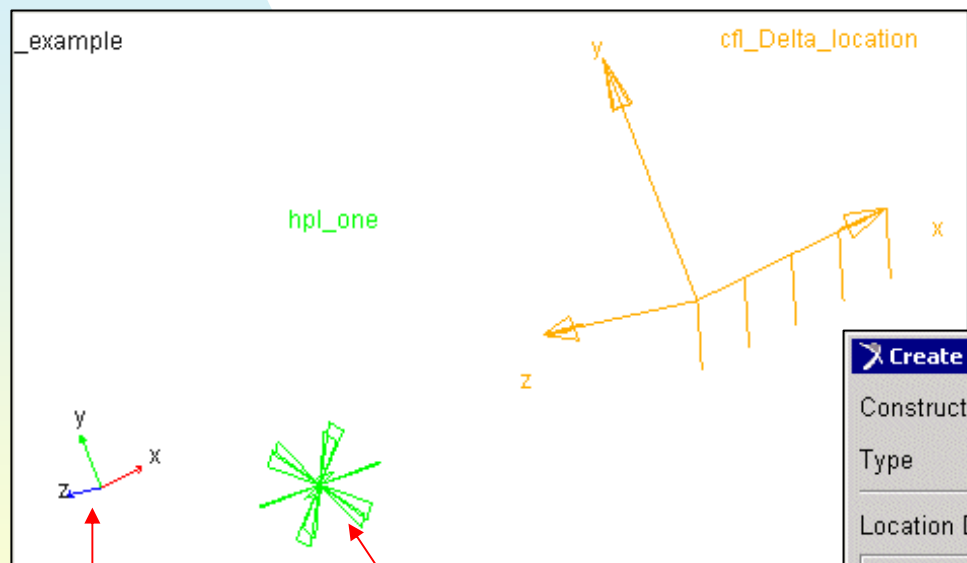


- 你同样可以修改构造坐标系，通过 Build → Construction Frame → Modify。在对话框中选择你要修改构造坐标系的名字，然后修改其位置和方向。

位置参数化

■ 相对坐标系下位置

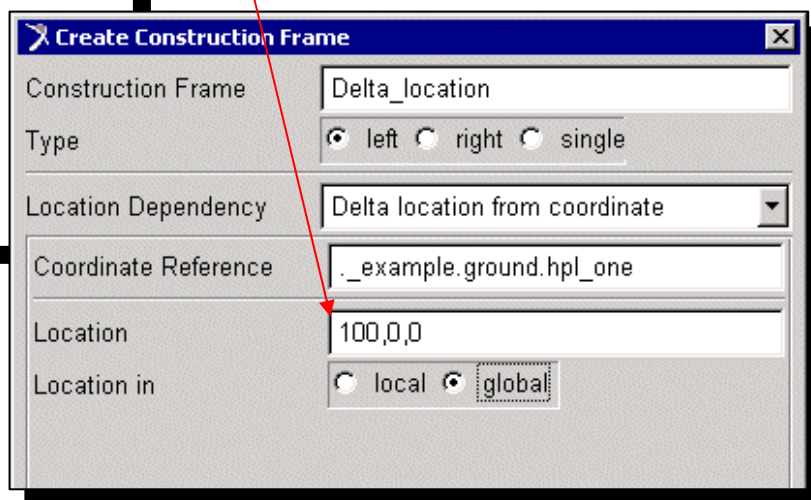
- ◆ 相对于一个预定义的参考坐标系的位置。
- ◆ 你可以定义在绝对的或相对的坐标下的 X, Y, Z 的位置。



位置在绝对坐标系 X 轴
100 个长度单位处

绝对坐标系

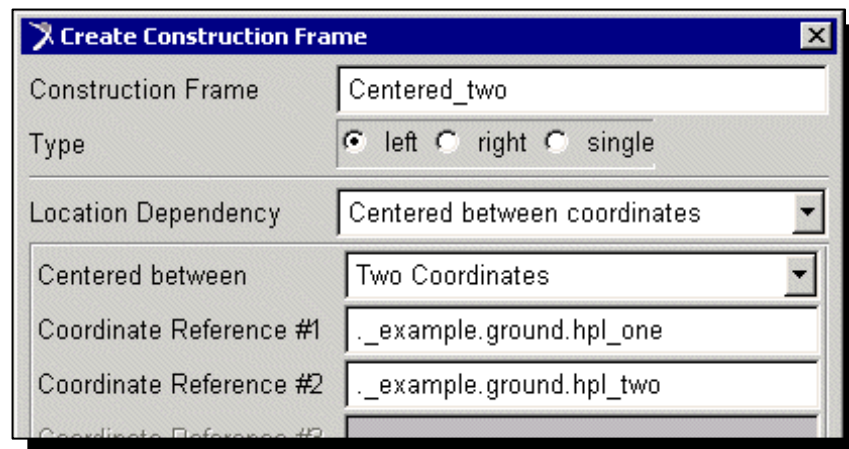
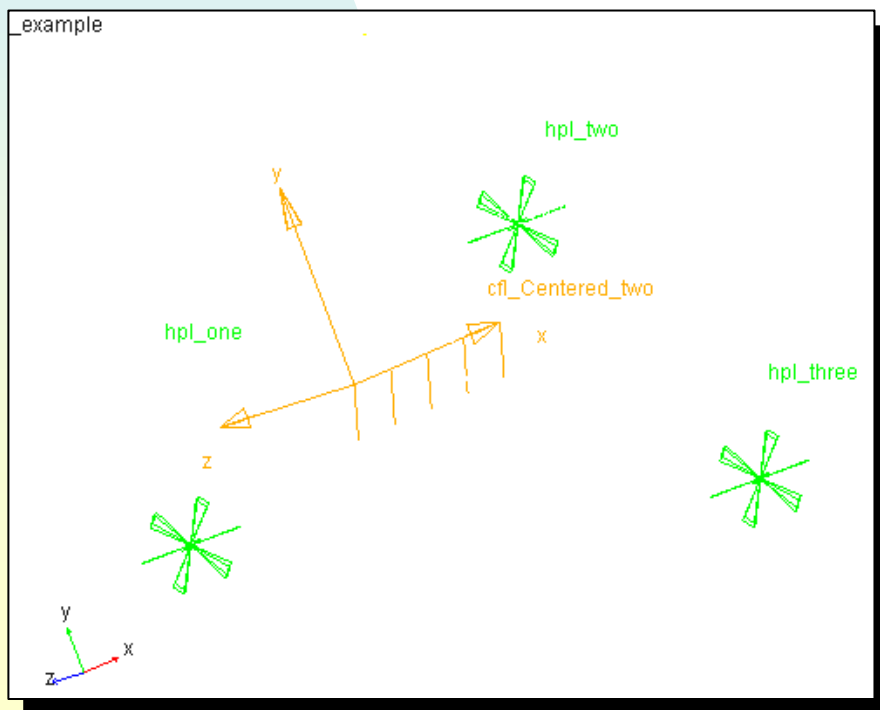
局部坐标系



位置参数化

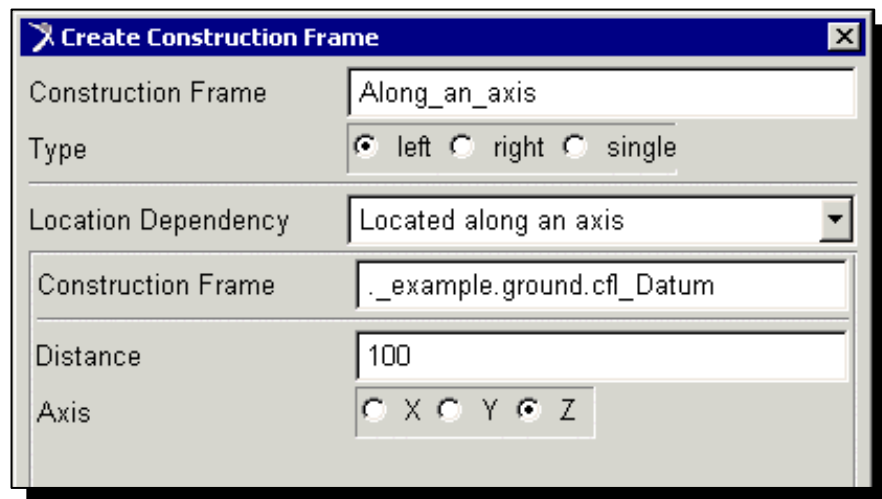
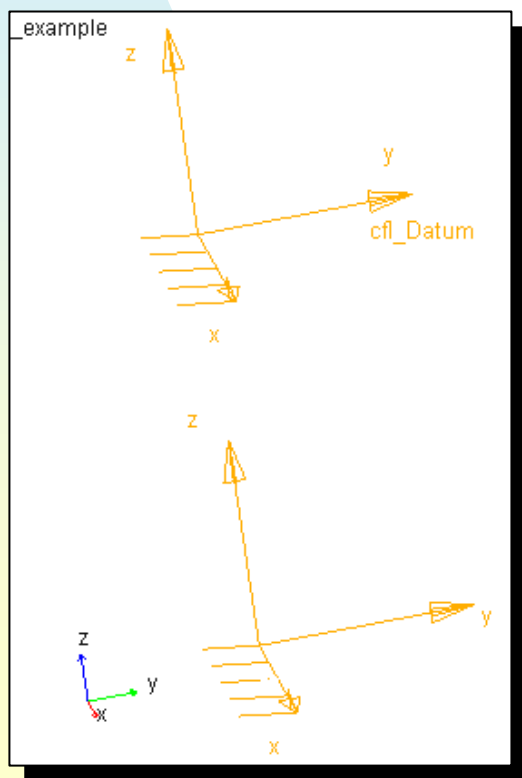
■ 在坐标系中间位置

- ◆ 使用两个坐标系的定义方法，在预先定义的两个坐标系假想的连线的中点处。
- ◆ 使用三个坐标系的定义方法，在预先定义的三个坐标系假想的三角形的几何中心处。



位置参数化

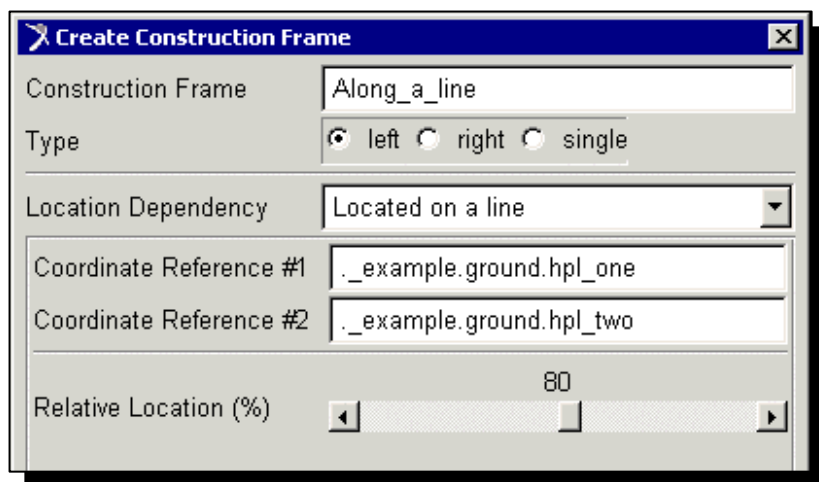
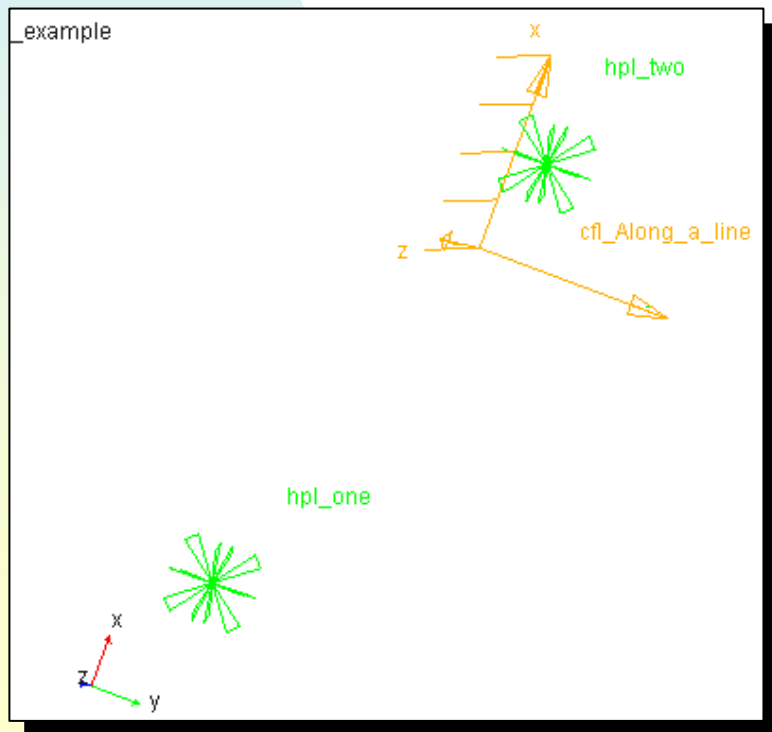
- 沿着一个坐标系的某一轴上
 - ◆ 此对象位于沿着一个坐标系的某一轴上的给定距离处。
 - ◆ 本例中，该对象位于沿着一个参考坐标系的 Z 轴上距离 100 单位长度处。



位置参数化

■ 在一条线上

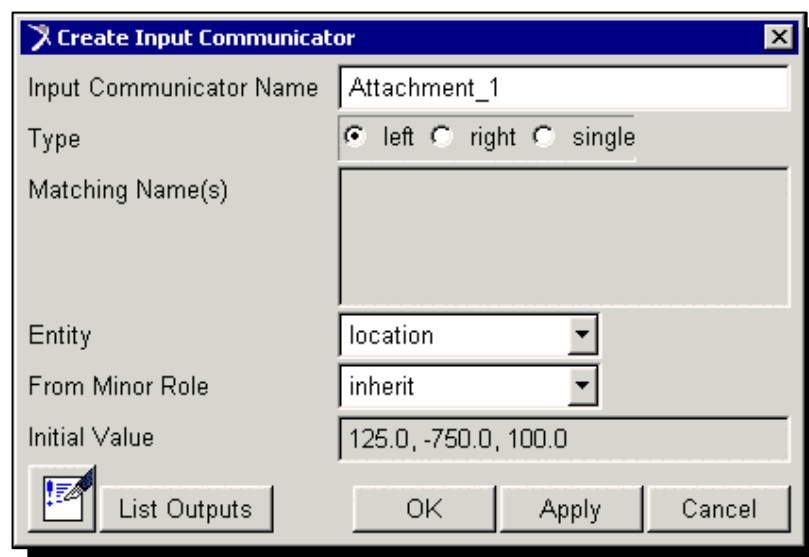
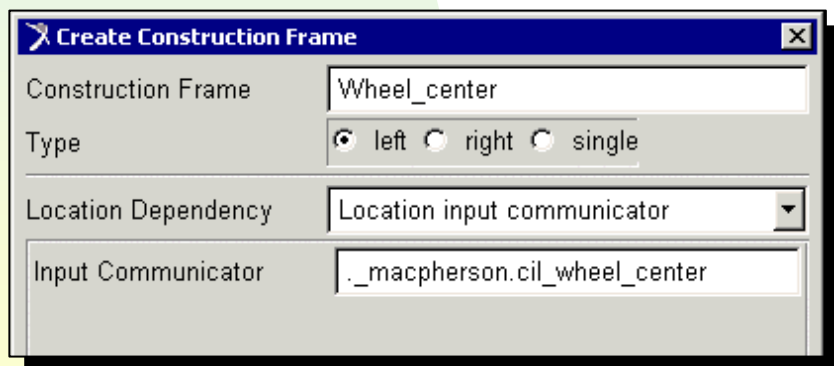
- ◆ 定位一个对象在两个参考坐标系的连线上。
- ◆ 距离按照从第一个参考坐标系到第二个参考坐标系之间的百分比为多少来定义。



位置参数化

■ 位置输入信息交换器

- ◆ 使用选择的输入信息交换器的位置数据来定位该对象。
- ◆ 使用此选项来定位一个对象，使其对应于其它模板下的参考坐标系。
- ◆ 在装配阶段，位置数据通过输入信息交换器从其它模板相对应的输出信息交换器传过来。依据那个点，确定该对象的初始位置在输入信息交换器所处的位置。

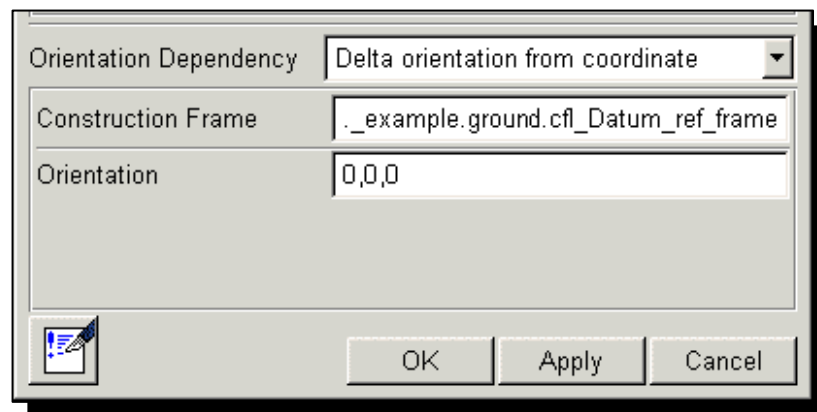
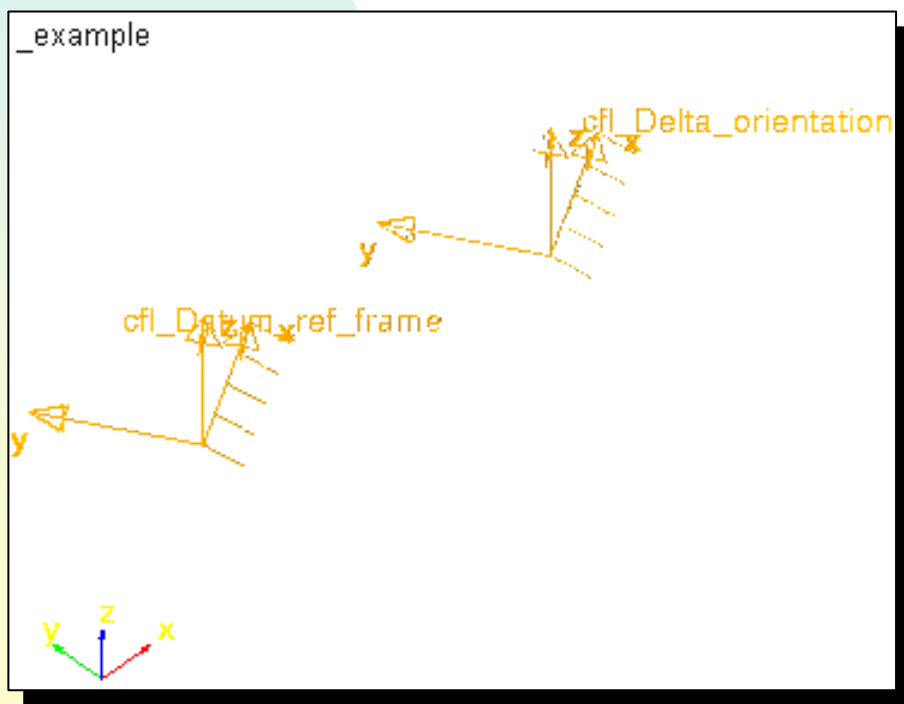


位置参数化

- 有关更多信息交换器方面的信息，参见下列的 KBAs：
 - ◆ Article 9182 -- Use of mount and location communicators at:
<http://support.adams.com/kb/faq.asp?ID=9182>
 - ◆ Article 9184 -- Example of use of location communicators at:
<http://support.adams.com/kb/faq.asp?ID=9184>

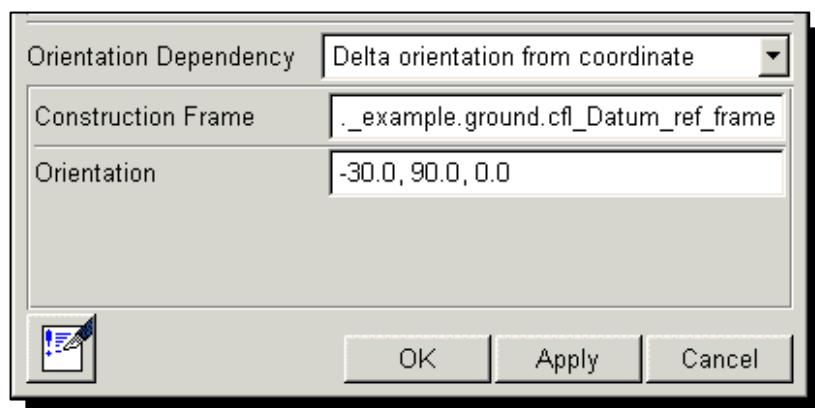
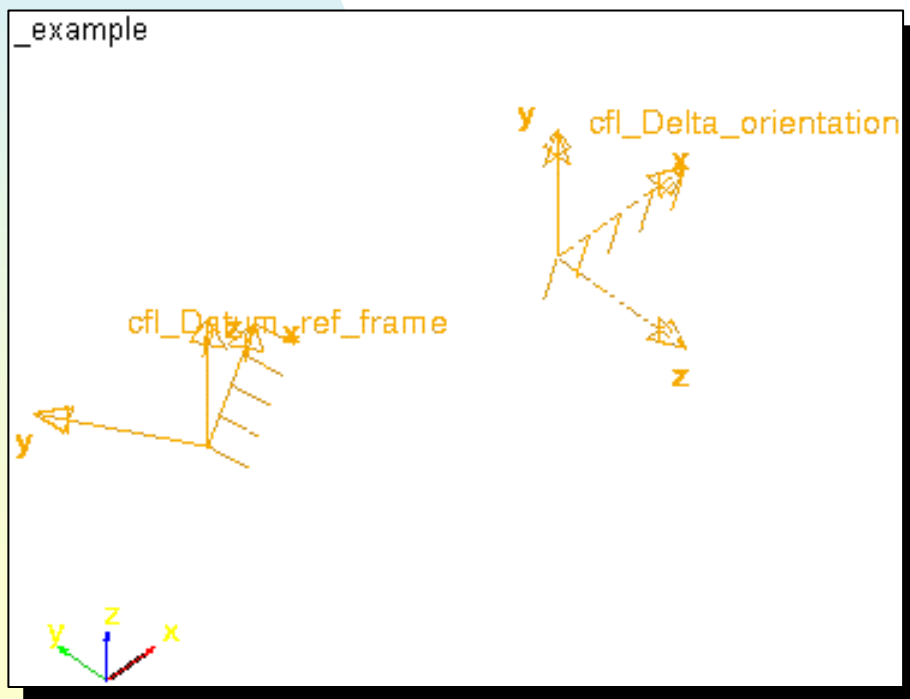
方向参数化

- 相对于某一个坐标系的坐标方位角
 - ◆ 相对于参考坐标系定向对象，使用欧拉方位角。
 - ◆ 在本例中，对象相对于参考坐标系在方位上一致。



方向参数化

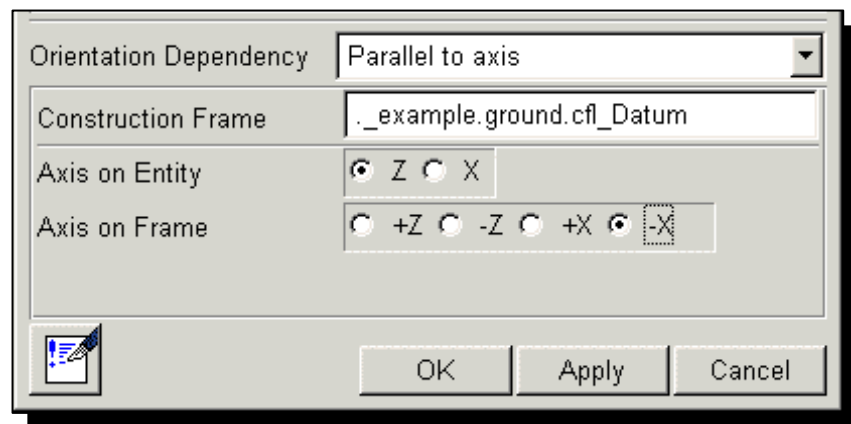
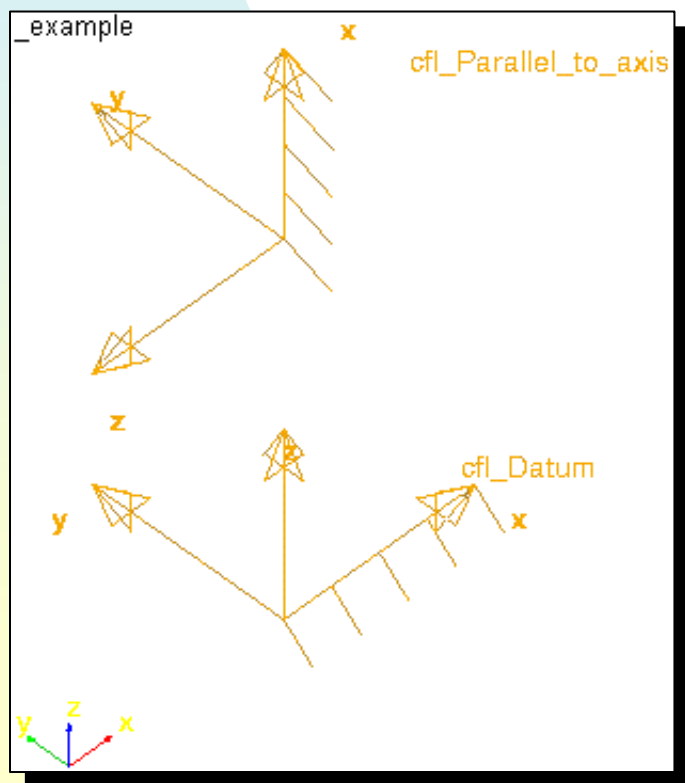
- 相对于参考坐标系的方向角
 - ◆ 相对于参考坐标系，给定欧拉角定义该对象的方位。
 - ◆ 在本例中，该对象相对于参考坐标系的 Z 轴先旋转 -30 度，再相对于新的 X 轴旋转 90 度。



方向参数化

■ 平行于另一轴

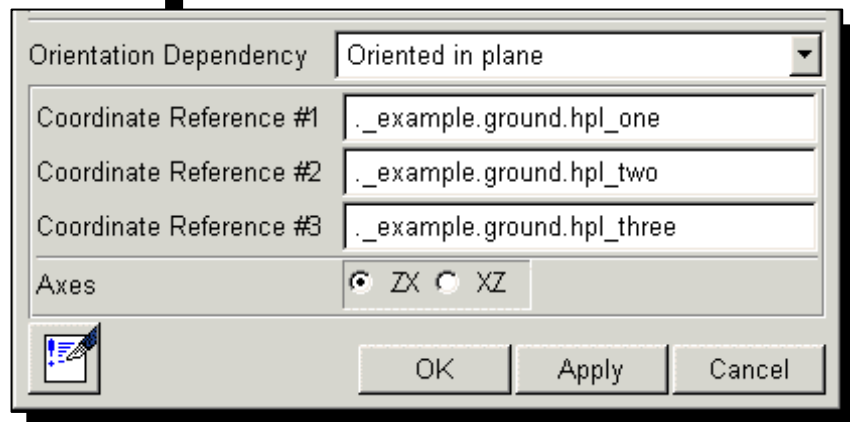
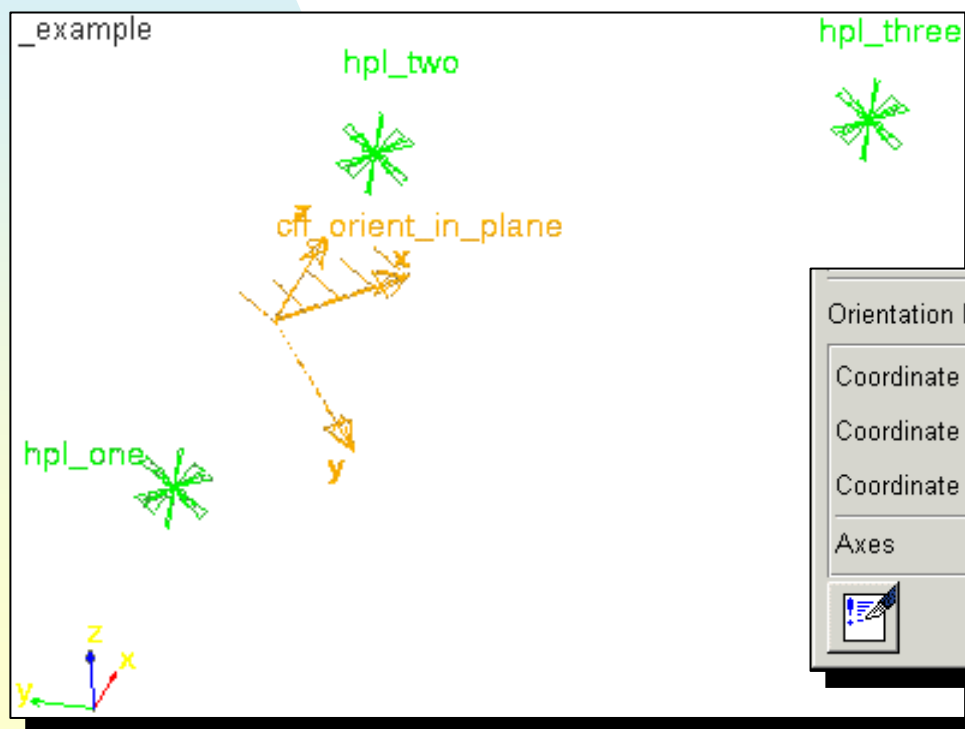
- ◆ 确定该对象的某一个轴与参考坐标系的某一个轴的方向平行。
- ◆ 在本例中，该对象的 z 轴被转到与参考坐标系的 $-x$ 轴方向平行。



方向参数化

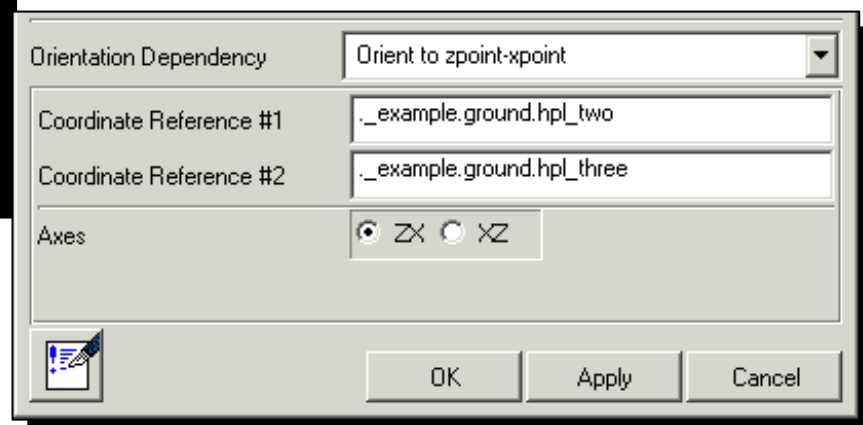
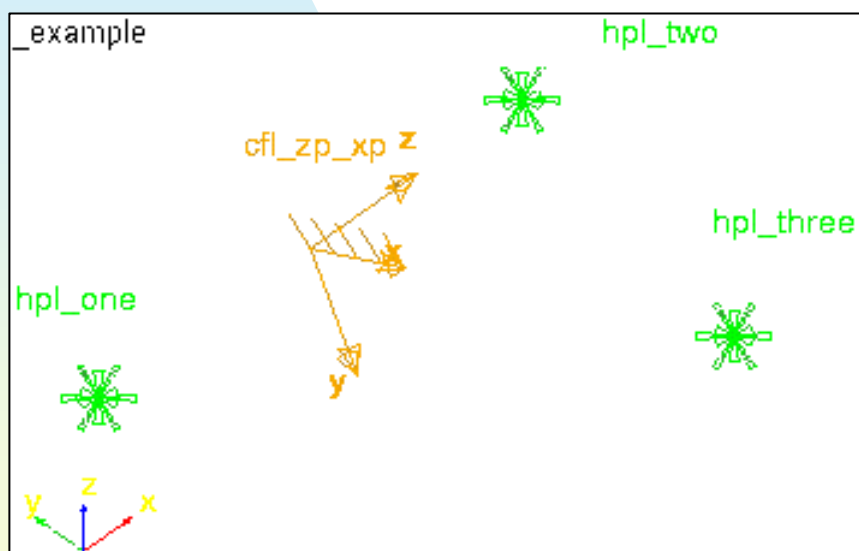
■ 由平面确定方位

- ◆ 使该对象的 z 轴沿着第一个和第二个参考坐标系的连线方向，再绕 Z 轴旋转该对象使其 ZX 平面与三个参考坐标系所构成的平面平行。



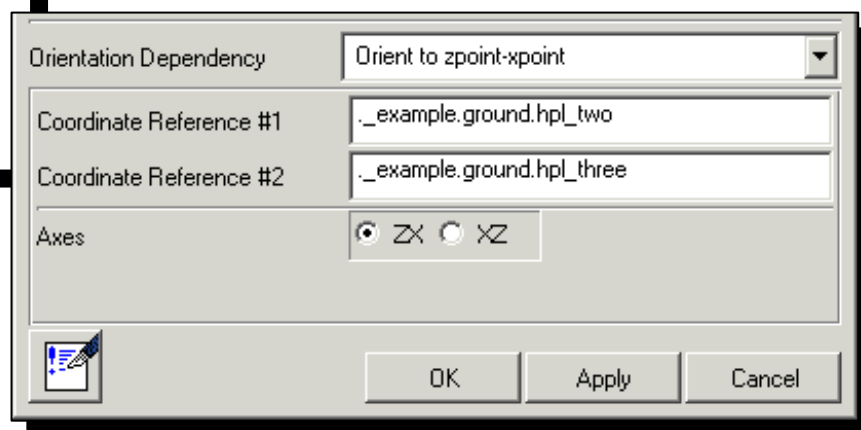
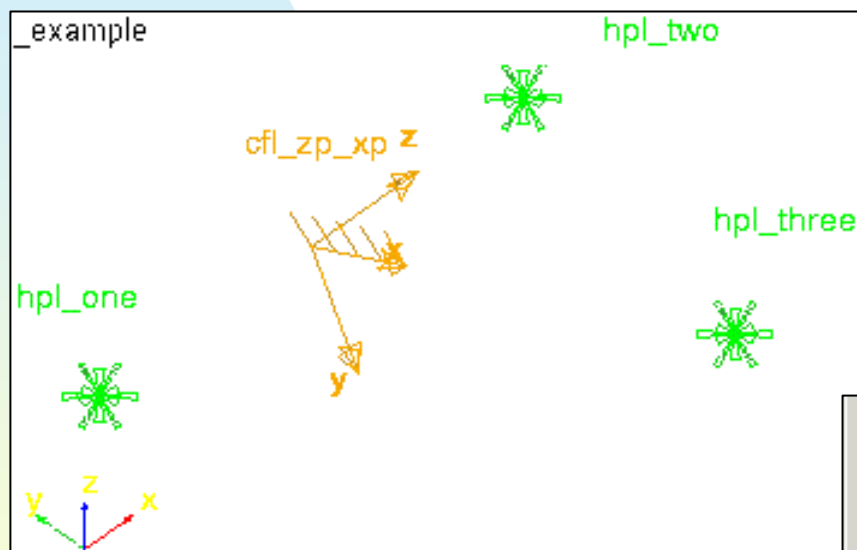
方向参数化

- 由 zpoint-xpoint 确定方位
 - ◆ 使 z 轴指向第一个参考坐标系，同时使 x 轴指向第二个参考坐标系。



方向参数化

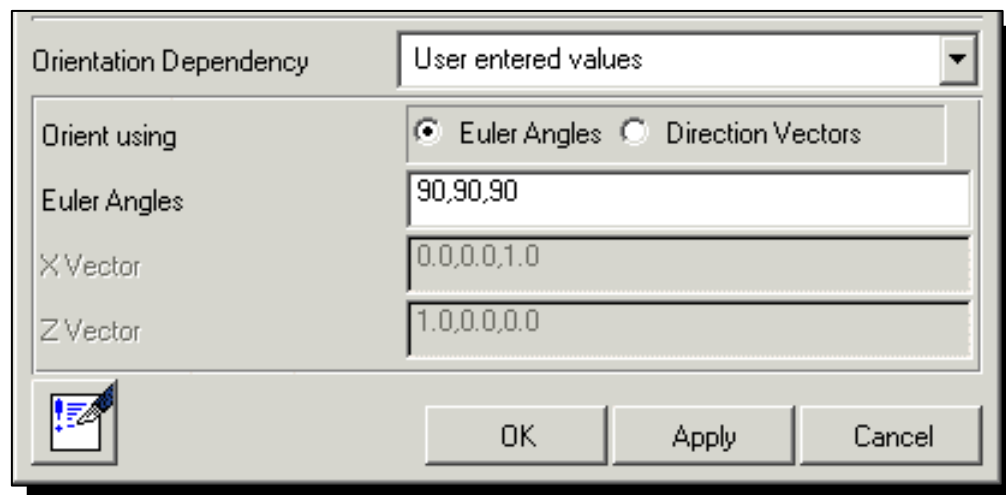
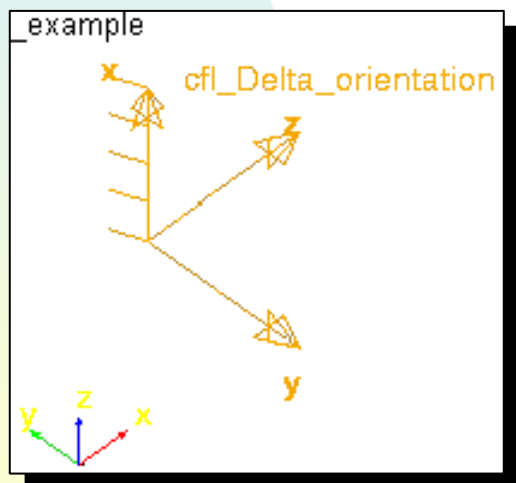
- ◆ 或者使 X 轴指向第一个参考坐标系，同时使 Z 轴指向第二个参考坐标系。



方向参数化

■ 用户直接输入方向角

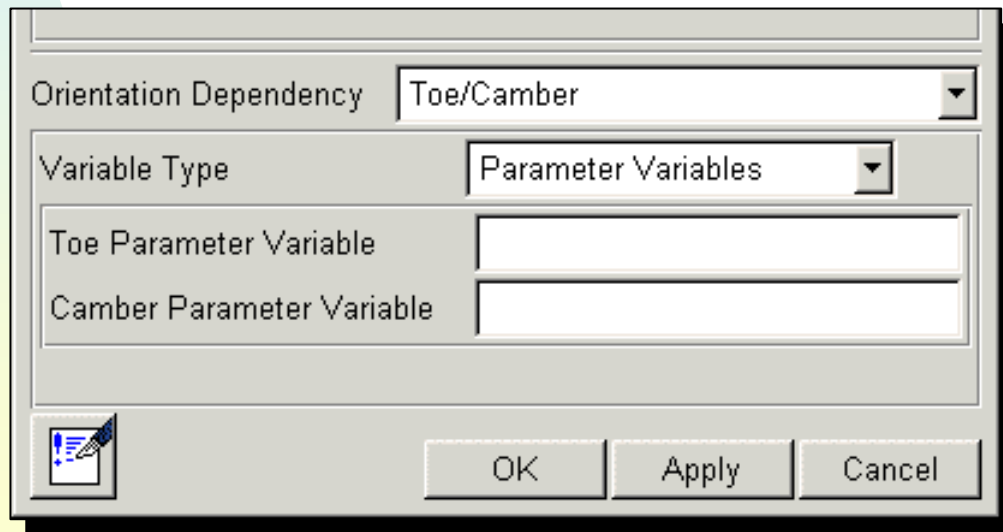
- ◆ 直接输入该对象直接相对于全局坐标系的三个欧拉角。



方向参数化

■ Toe/Camber

- ◆ 设置前束角(toe)和外倾角(camber)的方向。通常来讲，前束角(toe)和外倾角(camber)用来定义车轮中心旋转轴的绝对坐标系下的方向。你可以从参数变量或输入信息交换器中得到前束角(toe)和外倾角(camber)的数值。



The image shows a software dialog box titled "Orientation Dependency". It contains the following fields and controls:

- Orientation Dependency:** A dropdown menu currently showing "Toe/Camber".
- Variable Type:** A dropdown menu currently showing "Parameter Variables".
- Toe Parameter Variable:** An empty text input field.
- Camber Parameter Variable:** An empty text input field.
- Buttons:** "OK", "Apply", and "Cancel" buttons at the bottom right.
- Icon:** A small icon of a notepad and pencil is located at the bottom left of the dialog box.

第 10 章 建立模板

- 本章详细介绍如何建立模板，相应的一些元件以及在 ADAMS/Car 谱系下使用的目的。

建立模板

■ 本章内容：

- ◆ 模板综述
- ◆ 模板的拓扑关系
- ◆ 文件的构成
- ◆ 建立一个新的模板
- ◆ 部件的类型
- ◆ 刚体和弹性体（部件）
- ◆ 几何外形
- ◆ 连接方式 (约束和衬套)
- ◆ 弹簧
- ◆ 阻尼器
- ◆ 止挡和反向止挡
- ◆ 前束角(Toe)/外倾角(Camber)和悬挂参数数组
- ◆ 可调载荷
- ◆ 一些建议

模板综述

- ◆ ADAMS/Car 模板为参数化的模型，你可以利用模板定义车辆各组成之间的拓扑关系。建立一个模板，意味着你要定义部件，部件之间的连接关系以及与其它模板和试验台之间需要哪些信息交换器。一个模板可能代表一个元件或一组元件。
- ◆ 在模板建模阶段，准确的定义部件的质量特性或载荷的特性并不是特别重要，因为这些数据能够在子系统阶段重新设置。然而，正确的定义部件之间的连接关系以及要与其它模板之间需要交换哪些信息则是非常重要的，因为，你在子系统阶段不能再作修改。
- ◆ 当建立模板时，要注意整个系统的装配过程。亦即，确认你所建立的模板能够与其它的模板和你指定的试验台之间能够正确地进行信息的交换。
- ◆ 信息交换器定义不同的子系统之间需要交换哪些信息。我们将在第 11 章详细讨论信息交换器的问题。

模板的拓扑关系

- 在 ADAMS/Car 中，意味着生成一系列的元件的拓扑关系,包括硬点、部件、连接方式和定义子系统的参数，具体如下：
 - ◆ 生成硬点和构造坐标系 – 你首先要生成硬点和构造坐标系。硬点和构造坐标系为 ADAMS/Car 中的模型元件，主要用于定义模型中一些关键的位置和方向。它们是最常用的建模元件，主要用于参数化更高层次的对象。硬点的位置可以用来确定大部分部件及其连接点的位置。硬点只需要定义其位置，构造坐标系则需要定义其位置和方向。
 - ◆ 生成部件 – 你定义硬点和构造坐标系之后，你可以利用它们生成部件。

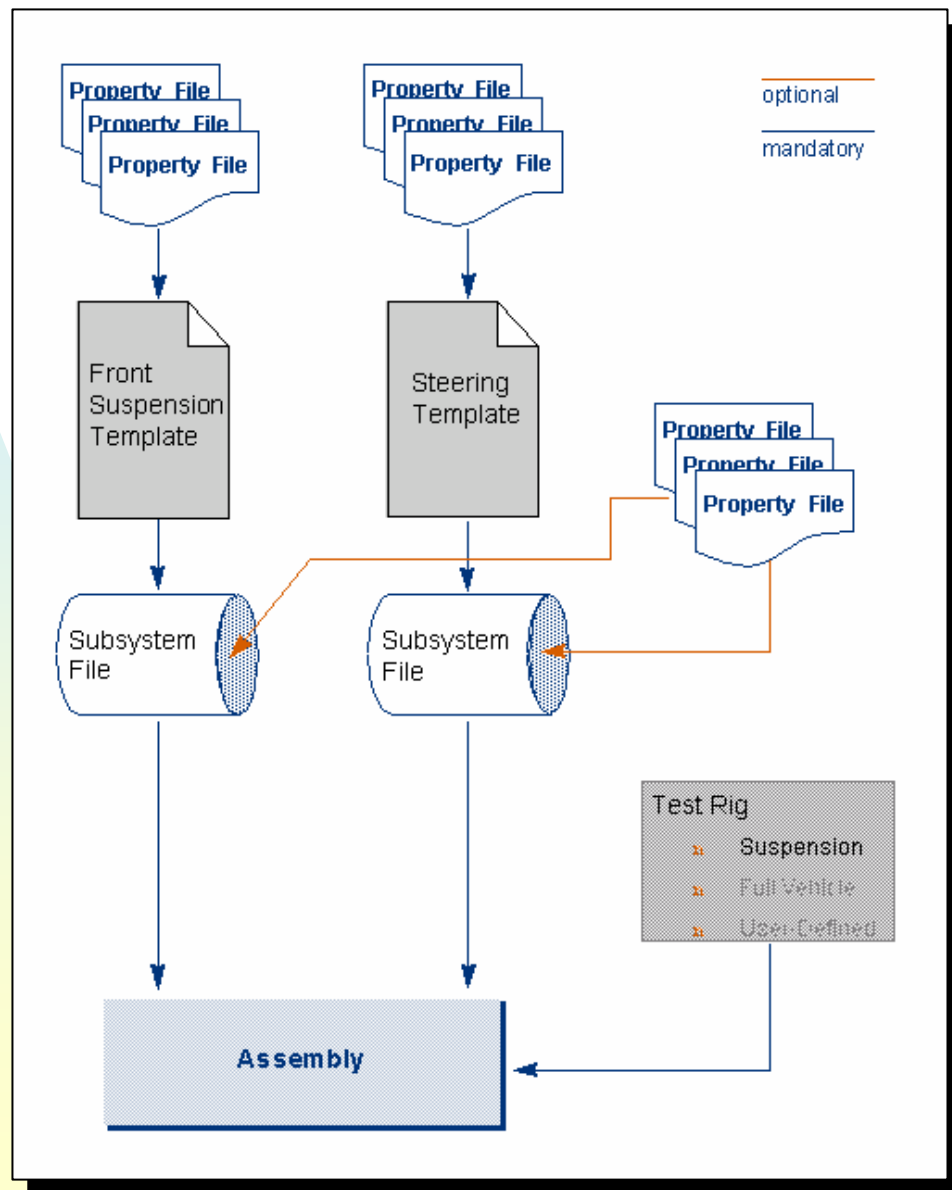
模板的拓扑关系

- ◆ 生成连接 – 最后，你需要生成连接，比如，约束、衬套和参数，主要用来告诉 ADAMS/Car 模型中的部件之间是如何相互作用的。你可以对于不同的分析类型定义不同的连接方式。柔顺性分析可以使用衬套，而运动学分析则使用约束。
- 在你开始定义你的模板之前，你必须确定你的模型中使用什么元件来定义模型最合适。同时你还需要确定什么样的几何外形似乎更合适作为你模型下部件或干脆就不选任何几何外形。一旦你确定后，你就可以生成模板以及模板下基本的拓扑关系了。

文件的构成

- ◆ MSC.ADAMS 的基于模板建模产品中相关的文件构成由存储在数据库中的下面的一些类型文件组成：
 - 特性文件
 - 模板文件
 - 子系统文件
 - 装配文件
- ◆ 下页图阐释 MSC.ADAMS 的基于模板建模产品是如何使用相关文件的。一个模板使用特性文件提供一些元件，如弹簧、阻尼器、衬套等的数据库。当生成一个新的子系统时，你可以使用原来模板中所参照的特性文件或使用其它数据库下的特性文件或具有不同的文件名字的特性文件，如虚线所示。一系列的子系统合并在一起构成一个装配。

文件的构成



建立一个新的模板

- ◆ 要建立一个新的模板，你将生成各种部件、几何外形、连接关系、约束以及其它所有生成你的模型所必须的元件。要建立一个模板，你需要首先了解 ADAMS/Car 中所有可以使用的元件，下面将做具体的描述。
- ◆ 在本节中将要描述的有：
 - 刚性体 (部件)
 - 几何外形
 - 连接关系 (约束和衬套)
 - 弹簧
 - 阻尼器
 - 止挡和反向止挡
 - 悬挂参数数组
- ◆ 模板中还需要定义信息交换器，为 ADAMS/Car 中一个非常重要的环节，通过信息交换器，你可以在子系统之间以及在装配时传递各种必要的信息。下节将重点描述信息交换器。

部件的类型

- 刚性体 (部件)
 - ◆ 为可以运动的部件。
 - ◆ 具有质量和转动惯量等力学性质。
 - ◆ 不能变形。
- 弹性体
 - ◆ 为可以运动的部件。
 - ◆ 具有质量和转动惯量等力学性质。
 - ◆ 受力时会产生变形。

部件的类型

■ 大地部件

- ◆ 每个模型中都必须存在的对象。
- ◆ 定义全局坐标系 (GCS) 和坐标系原点，因此，在任何时候都保持固定不动。
- ◆ 起着惯性坐标系的作用，用来计算其它运动部件的速度和加速度。

■ 装配部件和开关部件

- ◆ 装配部件为无质量的部件，将在装配过程中为其它的部件所替换。
- ◆ 开关部件也是无质量的部件，并起到连接的切换作用，类似于一个开关。通过改变开关部件，一个部件将与另外的部件相连接。

刚体和弹性体（部件）

■ 刚性体 (部件)

- ◆ 一个刚性体部件被称为一般部件，或简作 ge[lrs] (ge 的涵义是一般部件，[lrs] 为左、右或单个的意思)。这些部件可以相对于其它部件运动并具有下列属性：
 - 质量
 - 转动惯量
 - 初始位置和方向 [称作物体局部参考坐标系 (LBRF) 或部件坐标系(PCS)]
 - 初始速度
- ◆ 要在 ADAMS/Car 中生成一个 新的刚性体，可以从 Build -> Parts -> General Part，然后选择：
 - **New** – 指定物体局部参考坐标系 (LBRF) 的位置和方向，同时指定该部件的质量特性。从操作不生成任何部件的几何外形。
 - **Wizard** – 选择两个或三个硬点或构造坐标系，该部件将据此被参数化。该过程将生成部件和几何外形。
- ◆ ADAMS/Solver将对部件进行分析，而不是对部件的几何外形。有关几何外形部分，参见后面的片子中有所介绍。

刚体和弹性体（部件）


■ 弹性体 (部件)

- ◆ 你可以通过导入外部文件的方式生成弹性体。有关弹性体的更多信息，参见第 12 章 – 使用弹性体。

几何外形

- 在 MSC.ADAMS 的产品线上一个非常重要的方面就是几何外形的观念 (ADAMS/View 中的几何图形)。几何外形主要作用在于加强部件的可视化效果，使用下面一些特性来代表：
 - ◆ Length
 - ◆ Radius
 - ◆ Width
 - ◆ Thickness
- 相对于部件而言，对于分析而言，几何外形不是必须的，因此，你可以有一个没有几何外形的部件，但是。不可以有一个没有部件的几何外形。

几何外形

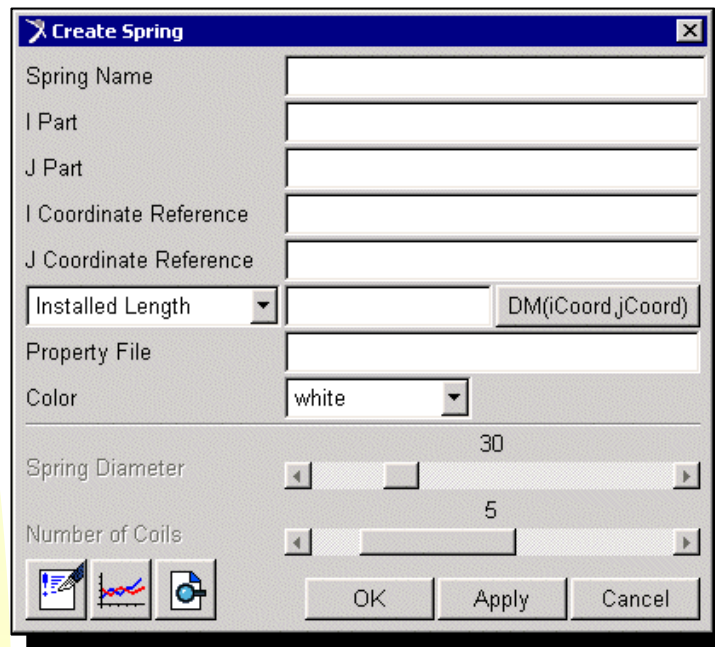
- 注意有时 ADAMS/Car 在你生成部件时自动地生成几何外形。例如：Wizard 选项就是这样。然而，如果你在生成部件时使用 New 选项时，就不会生成任何几何外形。你可以随后基于你自己的输入生成或增加几何外形。
- 同样需要注意的是基于几何外形计算的质量特性不是参数化的。亦即在几何外形出现改变的情况下，该部件的质量特性并不自动地改变。如果你想刷新一下改变几何外形的部件的质量特性，可以使用 Calculate Mass的工具  来重新计算一下。

连接方式 (约束和衬套)

- ◆ 连接方式，在 ADAMS/Car 中，意味着约束和衬套连接，并且这些意味着你的模型中各个部件之间是如何相互作用的。对于不同的分析类型，你可以定义不同的连接方式，如柔顺性分析和运动学分析模式。柔顺性分析模式使用衬套连接，而运动学分析模式使用约束连接。
 - 衬套连接(**Bushings**) – 提供部件之间三维的力和力矩连接，大小依刚度而定。你需要指定一些参数，如，刚度、预载荷以及偏置量来定义衬套。注意与 ADAMS/View 中的 BUSHING 卡片的不同之处。在 ADAMS/View 中，BUSHING 使用常数的刚度和阻尼系数。而在 ADAMS/Car 中 BUSHING 类似于 ADAMS/View 中的 FIELD。
 - 约束连接(**Joints**) – 提供部件之间运动学的约束关系。你需要指定你要在模型中添加的约束的类型。
- ◆ 你可以在一个模型中同样的两个部件之间既定义约束连接，也定义衬套连接，你可以在两种连接方式之间进行切换，激活一个，同时失效另外一个。这样的话，你可以使用同一个模型分别进行运动学和柔顺性的分析，你也可以进行混合的分析，即在一次分析过程中，既使用约束连接也使用衬套连接。

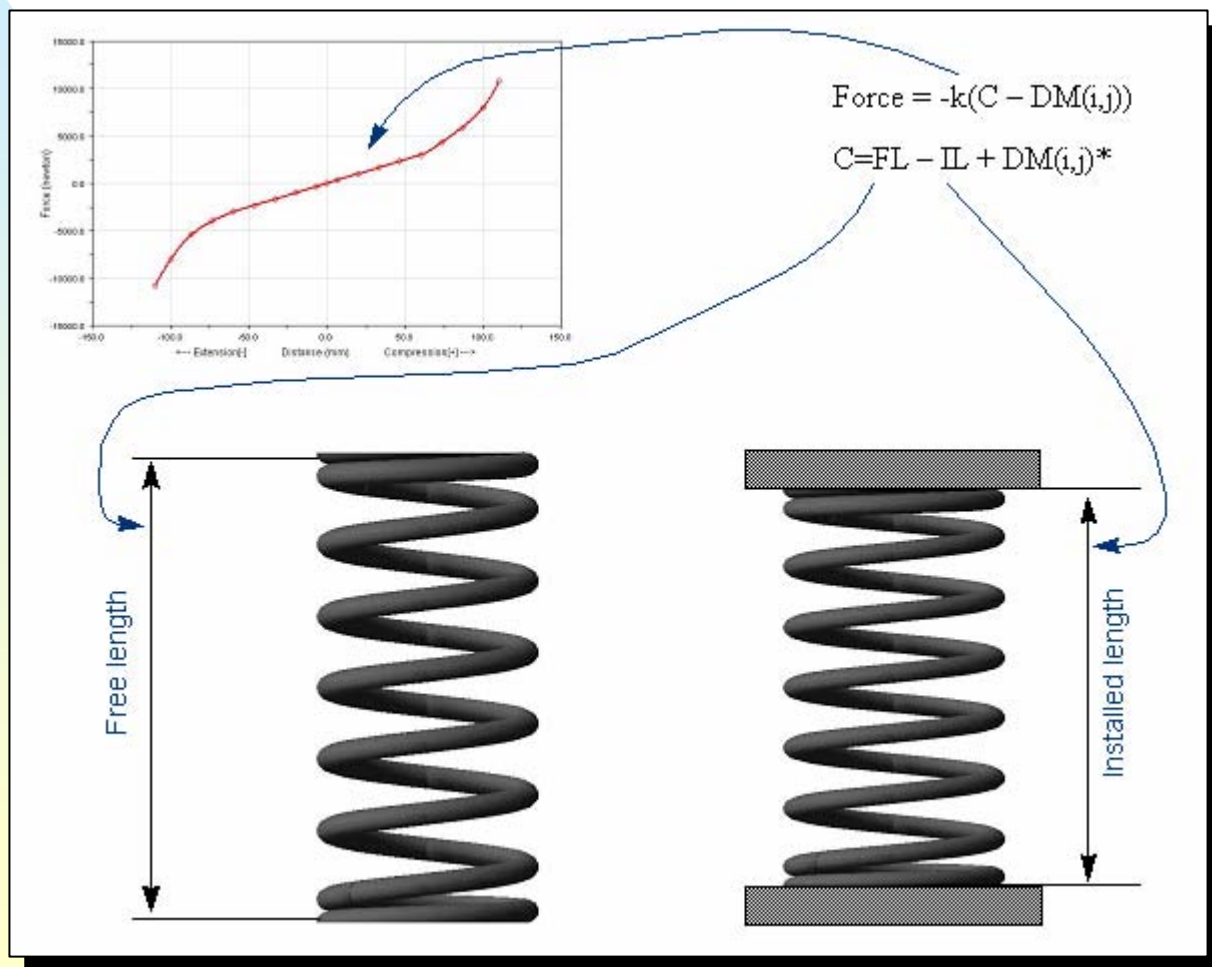
弹簧

- ◆ 当生成一个弹簧时，你需要两个坐标以定义弹簧的两个端点位置。坐标既可以是硬点，又可以是构造坐标系。然后，要定义弹簧，你需要指定以下参数：
 - 弹簧力作用的两个物体以及两个参考坐标系 (弹簧在两个物体上的连接点的位置)。
 - 弹簧的安装长度，被用来计算弹簧上的预载荷的大小。
 - 特性文件，包含弹簧的自由长度信息以及载荷/变形关系。



弹簧

- ADAMS/Car按照下图中的公式计算弹簧力的大小：



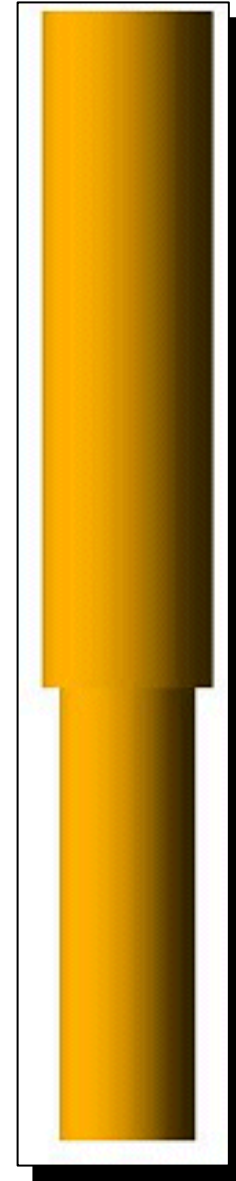
弹簧

■ 其中：

- ◆ C 为一个常数。
- ◆ FL 为弹簧的自由长，在弹簧的特性文件中定义。
- ◆ IL 为弹簧的安装长度。
- ◆ $DM(i,j)^*$ 为弹簧的 I 和 J 参考坐标系之间的初始位移。如果输入 $DM(i,j)^*$ 的值较小的话，ADAMS/Car 计算出来的弹簧的预载荷增加，反之，则减小预载荷。
- ◆ Force 代表弹簧载荷。
- ◆ K 为从弹簧的特性文件中导出的非线性弹簧的刚度。

阻尼器

- 在生成阻尼器时，你需要指定两个端点的位置和阻尼器的特性文件。与弹簧不同，阻尼器不需要预载荷。ADAMS/Car 同样生成阻尼器的几何形状，如右图所示。



止挡和反向止挡

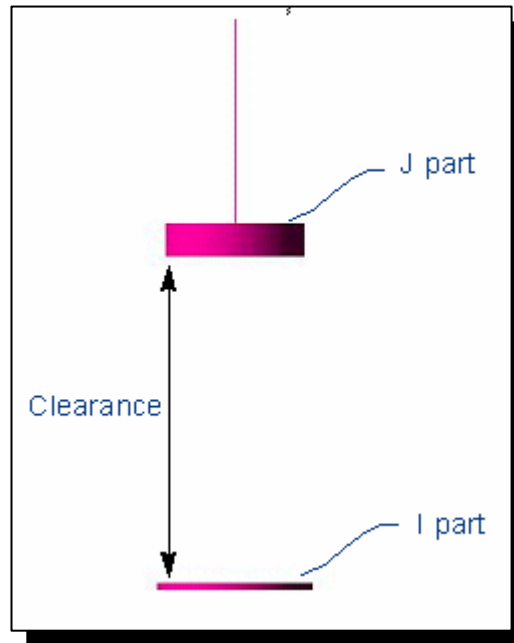
■ 止挡

- ◆ 定义两个部件之间力-位移的关系曲线。止挡力作用在两个部件上的用户指定的坐标系处，大小遵从所指定的特性文件中的力-位移的关系曲线。当两个坐标系标记点之间的距离超过一个给定的数值时止挡力起作用，止挡力起作用距离的定义有下面两种方式：
 - **Clearance** – 定义部件 I 向部件 J 运动多少距离时止挡力起作用。
 - **Impact length** – 定义部件 I 和部件 J 之间距离为多少时止挡力起作用。

止挡和反向止挡

■ 反向止挡

- ◆ 与止挡起作用的方式类似，但起反弹的作用，而不是像止挡那样跳动。ADAMS/Car 同时生成该元件的图例如下：



前束角(Toe)/外倾角(Camber)和悬挂参数数组

- ◆ 你可以设置悬架的前束角(toe) 和外倾角 (camber) ，通过选择 Build -> Suspension Parameters -> Toe/Camber -> Set。
- ◆ 在车辆的设计位置，悬架的前束角(toe) 和外倾角 (camber)可能不是 0。你必须将悬架的这些参数从选架子系统传递到车轮子系统。此处你设置设计位置的前束角(toe) 和外倾角 (camber) 的数值后，ADAMS/Car 自动地生成参数变量(parameter variables) 和输出信息交换器，你可以使用来定向车轮或其它对象。
- ◆ 你首先生成变量定义前束角(toe) 和外倾角 (camber) 。因为这些变量为悬架分析常用的参数。ADAMS/Car 一次性地生成这些参数。使用转向轴来定义，悬架参数数组将计算。
- ◆ 要打开定义悬架参数数组定义的对话框，可以选择 Build -> Suspension Parameters -> Characteristic Array -> Set。

前束角(Toe)/外倾角(Camber)和悬挂参数数组

- ◆ 你可以使用下面的两个方法之一生成转向轴：
 - 几何法 (**Geometric method**) - 选择两个不同部件上的两个点来定义转向轴。
 - 定轴法 (**Instant-axis method**) - ADAMS/Car 先索定弹簧的运动并施加递增的转向力矩或力，然后，从得到的车轮承载部件上的移动和转动量，ADAMS/Car 计算每个车轮承载部件的转动定轴。转动定轴即为转向轴。
- ◆ 最后，你需要设置悬架的类型是独立悬架或非独立悬架。
- ◆ Steering axis -主销轴

可调载荷

- ◆ 可调载荷为特殊的模板建模用户定义的元素 (UDE)。你可以使用可调载荷来适应各种条件，以满足你的模型中静力学参数的要求。比如，如果你想要设置一根杆的长度在静力学分析时为一个特定的数值，则可调载荷将会自动地改变，直至满足所需要的条件。你可以通过选择 Build -> Adjustable -> Force New 定义可调载荷。
- ◆ 在 ADAMS/Car 中，举个例子来说，一个使用可调载荷典型的应用就是在静力学分析时设置前束角(toe) 和外倾角 (camber) 的大小。你可能使用两个部件来定义系杆并使用移动副定义其连接关系，然后定义两个部件之间的可调载荷以设置前束角(toe) 和外倾角 (camber)。如果不使用可调载荷，当车辆达到静平衡位置时，前束角(toe) 和外倾角 (camber) 可能并不是你要的值。你使用可调载荷来定义前束角 (toe) 和外倾角 (camber)在静平衡位置的值。

可调载荷

- ◆ 现在生成一个一元力为作用在两个部件之间的载荷。可调载荷作用在两个部件之间并在静力学分析过程中不断的调整，以使当前计算得到的前束角(toe) 和外倾角 (camber) 与要求的前束角(toe) 和外倾角 (camber) 相一致。
- ◆ 在ADAMS/Car 的共享数据库下的模板 _double_wishbone_torsion中包含一个可调载荷的例子可供参考。

一些建议

- ◆ 先在纸上划一下草图。
- ◆ 开始总是先定义硬点，并给其赋予合适的名字和数值。这样可以保证至少在你们公司内部你所建立的模板能够为其它设计人员成功地使用。
- ◆ 可以按照 Build 菜单顺序进行。
- ◆ 在生成一个新的对象时尽可能的增加一些评述注释。
- ◆ 对某些元件，在修改对话框内比在生成对话框内有更多的选项(比如刚性体)。在这种情况下，可以先生成这些元件，再使用修改对话框修改这些元件。

一些建议（续）

- ◆ 部件可以为：刚性体/弹性体、装配部件、开关部件、非线性杆件(等效于 ADAMS/View 中的离散梁)。
- ◆ 刚性体的质量特性可以采用用户输入的方式或基于几何外形的方式进行定义。
- ◆ 支持的几何外形包括：圆柱、球、arms、连杆和轮廓线等。
- ◆ 连接方式可以是运动学的约束或动力学的衬套。
- ◆ 力元件包括：弹簧、阻尼器、止挡、可调载荷等等。
- ◆ 一些高级的元件将在后面的章节中介绍。



你可以通过按 F1 键显示帮助的对话框。

练习 9 模板建模入门

- 练习指导 *Getting Started Using ADAMS/Car* 中的模板建模入门部分。
- 在练习过程中，要注意，在生成新的模板时，注意增加注释，否则，别人很难使用你的模板。



本练习大约需要两个小时完成。

第 11 章 信息交换器

- 本章介绍信息交换器，在 ADAMS/Car 中，信息交换器主要用来控制装配过程以及子系统之间需要交换哪些信息。

信息交换器

■ 本章内容：

- ◆ 信息交换器的种类
- ◆ 信息交换器的分类
- ◆ 信息交换器的几何对称性
- ◆ 信息交换器的角色
- ◆ 信息交换器的命名
- ◆ 在装配过程中信息交换器的匹配
- ◆ 与试验台之间信息交换器的匹配

信息交换器的种类

- 信息交换器是 ADAMS/Car 中的关键元件，使得不同的子系统(包括试验台)之间在组装成装配时可以相互交换信息。
- 一个信息交换器为 ADAMS/View 中的变量，包含：
 - ◆ 对象 (例如，一个部件、变量、坐标系标记点或约束)
 - ◆ 实数值 (例如，x、y、z 的位置)
 - ◆ 字符串

信息交换器的种类

■ 信息交换器的种类

- ◆ 一个装配需要在其子系统之间双向的传递一些数据，要实现这种双向的数据传递，ADAMS/Car 提供两种类型的信息交换器：
 - 输入信息交换器(**Input communicators**) – 需要来自其它子系统或试验台的信息。
 - 输出信息交换器(**Output communicators**) – 向其它子系统或试验台提供的信息。
- ◆ 可以将输入信息交换器看作是一个电插头，而输出信息交换器则可以看作是一个电源插座。电插头需要从电源插座取电。
- ◆ 例如，一个装配信息交换器For example, a mount communicator in the rack and pinion steering templates outputs the rack part name so that tie rods of suspension templates can attach to the rack. In addition, a mount communicator in the steering template inputs a part name to determine where to attach the steering column to the body.

信息交换器的分类

- 信息交换器的分类表示它所交换的信息的类型。例如，一个硬点类型的信息交换器将通过硬点的名字和部件的名字传递位置。信息交换器的分类和每类信息交换器所传递的信息如下表所示。分类既适用于输入信息交换器也适用于输出信息交换器。

信息交换器的分类

分类：	交换的信息：
装配部件	各子装配之间连接的部件名称。缺省地，ADAMS/Car 在生成装配部件的同时，自动地生成一个输入装配部件信息交换器。
位置	硬点或构造坐标系的位置。如果硬点或构造坐标系为几何对称的，ADAMS/Car 将为对称的对象分别生成一个输入信息交换器。
坐标系标记点	构造坐标系和部件名称，提供位置、方向以及部件的信息。如果构造坐标系为几何对称的，ADAMS/Car 将为对称的对象分别生成一个输入信息交换器。
约束	约束的名称
带驱动的约束	约束的名称
衬套	衬套的名称
数组	ADAMS/Solver 数组的名称
样条数据	样条数据的名称

信息交换器的分类

微分方程	微分方程的名称
Solver 变量	ADAMS/Solver 变量的名称。你用的必须是 ADAMS/Solver 的变量而不是 ADAMS/View 的变量。与 ADAMS/View 的变量不同之处在于，ADAMS/Solver 的变量在分析过程中要时时的计算。ADAMS/Car 生成 ADAMS/Solver 的变量为状态变量。
驱动	驱动的名称
部件	部件的名称
方向	构造坐标系的方向
实参数	实数类型的参数变量的名称
整参数	整数类型的参数变量的名称
载荷	载荷的名称

注意：在对话框中，术语 *class* 和 *entity* 是可以相替换的。

信息交换器的几何对称性

- 信息交换器可以是单个的，也可以是左右成对的。但下列类别的信息交换器，如数组、微分方程、驱动、参数变量、Solver 变量和样条数据等则为非几何对称的，因此，缺省地，它们始终为单个的。

*注意：*在生成装配部件时自动生成的输入装配部件信息交换器自动地继承所参考的坐标系标记点的几何对称性决定其几何对称。

信息交换器的角色

- 每个信息交换器都需要有一个次要角色。次要角色定义信息交换器在装配中的位置。ADAMS/Car 提供五种缺省的次要角色：
 - ◆ Front
 - ◆ Rear
 - ◆ Trailer
 - ◆ Inherit
 - ◆ Any
- 如果你选择 *inherit* , 表明信息交换器的次要角色将继承使用该模板的子系统的次要角色。

信息交换器的角色

- 你可以在生成信息交换器时定义其次要角色。例如，如果你想从其它指定角色的子系统提供输入或输出，那么你可以在生成信息交换器时设置其次要角色。但我们推荐，你并不设置信息交换器的其次要角色，相反，由子系统自行决定其次要角色。例如，一个悬架的模板可能既用于定义前悬架子系统也用于定义后悬架子系统，通过子系统自行决定其次要角色，装配过程会将转向系统与前悬架系统相连，而不会与后悬架系统相连。

信息交换器的命名

- 在你生成一个信息交换器后，ADAMS/Car 会在其名字的前面分配一个前缀。例如，前缀 *ciI_*，其中：*ci* 代表输入信息交换器，如果是输出信息交换器，则为 *co*。*I* 代表其为一对称信息交换器的左侧那个。如果是右侧，则 ADAMS/Car 将使用 *r* (*cir*)，如果是单个的则为 *s* (*cis*)。

信息交换器的命名

- 如果你生成一个装配部件，ADAMS/Car 将自动地生成类别为 *mount* 的输入信息交换器，并使用装配部件的名字并加上前缀 *ci[lrs]_*，至于加什么视其是左、右还是单个的而定。例如，你生成一个装配部件的名称为 *mtl_rack_mount*，ADAMS/Car 将生成输入信息交换器的名称为 *ci_l_rack_mount*，其中，*l* 表示其为左侧的。注意：你不能生成一个装配输入信息交换器，你必须生成一个装配部件，ADAMS/Car 将自动地生成对应的信息交换器。

信息交换器的命名

- 在你给信息交换器命名的时候，你应该保证任何需要进行信息交换的输入输出信息交换器的名字是相匹配的。举例来说，你给在装配过程需要交换部件名称的信息交换器的名字可能为 *ci[lrs]_strut_mount* 和 *co[lrs]_strut_mount*。另外，如果你是使用 MSC.Software 的基于模板类的产品，你同样必须保证该产品的命名规则。

在装配过程中信息交换器的匹配

- 对在装配过程中需要交换信息的一对信息交换器而言，其名称上必须满足以下几点：
 1. 一致的相匹配的名字
 2. 具有完全相反的类型 (一个为输入，另一个则为输出)
 3. 必须具有相同的几何对称性 (左、右或单个)
 4. 具有相同的类别 (交换的信息类别相同)；例如，装配部件
 5. 具有同样的次要角色或被赋予 *any* 的次要角色

在装配过程中信息交换器的匹配

- 如果一个输入信息交换器没有对应的输出信息交换器，ADAMS/Car 将返回一条警告的信息，并且如果输入信息交换器的类别为 *mount* 的话，ADAMS/Car 将分配该装配部件到地球上。ADAMS/Car 返回一条警告的信息，因为你的输入信息交换器没有所需要对应的信息，因此，你的装配可能没有提供其正确工作所需要的所有的信息。

在装配过程中信息交换器的匹配

- 另一方面，如果输出的信息交换器没有链接到一个或多个输入信息交换器上，在装配过程中你将不会得到警告信息，因为输出信息交换器只是简单的发布一些信息，不会影响你的装配过程。
- 即使模型中有信息交换器不匹配，你仍然可以进行分析。事实上，你会发现这一点非常有用，尤其是在你只想分析一个子系统，而不包含其它子系统时。

在装配过程中信息交换器的匹配

- 例子，下列输入输出信息交换器在装配过程中匹配并交换部件的名称。

The pair:	Belongs to the class:	From minor role:	To minor role:
cil_strut_mount	mount	front	
col_strut_mount	mount		front
cil_strut_mount	mount	any	
col_strut_mount	mount		front
cil_strut_mount	mount	front	
col_strut_mount	mount		any

Diagram annotations:

- 1: Points to the first row of the table.
- 2: Points to the first two rows of the table.
- 3: Points to the first column of the table.
- 4: Points to the 'mount' entries in the second and third rows of the 'Belongs to the class' column.
- 5: Points to the 'front' entry in the first row of the 'From minor role' column.

在装配过程中信息交换器的匹配

- 另外，一个输入信息交换器只能跟一个输出信息交换器相匹配，而一个输出信息交换器可以跟多个输入信息交换器相匹配，这是因为输入信息交换器需要来自输出信息交换器的全部信息，而如果有多个输出信息交换器的话，输入信息交换器则无从判断并选择哪一个。

在装配过程中信息交换器的匹配

- 输出信息交换器只是发布一些信息，任何需要这些信息的输入信息交换器都可以使用。在装配过程中，你需要检查是否有警告信息提示，尤其是 *mount* 类别的输入信息交换器，没有给分配，因此将与大地相连接。

与试验台之间信息交换器的匹配

- 当你生成模板时，你必须满足下述条件以保证你新的模板能够正常工作：
 - ◆ 模板能够与其它模板和试验台兼容，比如试验台：.__MDI_SUSPENSION_TESTRIG，模板中必须提供正确的输出信息交换器。
 - ◆ 如果模板是一个悬架的模板 (例如，其主要角色为 suspension)，则该模板中必须包含悬架参数数组。悬架参数数组在悬架分析中表示主销轴线是如何计算的以及悬架是非独立的还是独立的等。

与试验台之间信息交换器的匹配

- 例子，要使悬架模板与悬架试验台相兼容的话，悬架模板必须包含下列输出信息交换器(不算其它的)：

The communicator:	Belongs to the class:	From minor role:	Matching name:
co[lr]_suspension_mount	mount	inherit	suspension_mount
co[lr]_suspension_upright	mount	inherit	suspension_upright
co[lr]_wheel_center	location	inherit	wheel_center
co[lr]_toe_angle	parameter_real	inherit	toe_center
co[lr]_camber_angle	parameter_real	inherit	camber_angle

与试验台之间信息交换器的匹配

- co[lr]_suspension_mount 输出信息交换器发布试验台的车轮要连接的映像部件 (通常为 hub/spindle 部件)
- co[lr]_suspension_upright 输出信息交换器则发布悬架系统中的向上部件，这样，垂直原始约束才能够正确地加到向上部件和映像部件 (例如：hub 部件) 之间，以便在进行静态仿真过程中将车轮的运动锁定。

与试验台之间信息交换器的匹配

- `co[lr]_wheel_center` output输出信息交换器则发布车轮的中心位置给试验台，这样，试验台才能够确定其相对于悬架的位置。在你生成这些类型的输出信息交换器时，一定要确认其次要角色设置为继承（inherit）。
- `toe` 和 `camber` 输出信息交换器(`co[lr]_toe_angle` and `co[lr]_camber_angle`) 向试验台发布悬架上的前束角和外倾角参数，这样的话，试验台才能正确地对车轮定向。



欲知试验台所需要的所有的信息交换器，请参阅 ADAMS/Car 在线帮助中 Templates tab 部分。

练习 10 取得信息交换器的信息

- 在本练习中，你将学习如何进行检查并显示信息交换器的信息，以帮助你更好地理解信息交换器。

练习 10 取得信息交换器的信息

■ 打开一个模板

要打开一个模板：

1. 从菜单 **File** 下选择 **Open**。
2. 在 **Template Name** 文本栏内点击鼠标右键并指向 **Search**，再选择 **<acar_shared>\templates.tbl**。
3. 双击 **rack_pinion_steering.tpl**。
4. 选择 **OK**。

齿轮齿条转向模型显示在你的窗口内。

练习 10 取得信息交换器的信息

■ 取得信息交换器的信息

要取得你的模板中信息交换器的信息：

- ◆ 从菜单 **Build** 下指向 **Communicator**，再选择 **Info**。

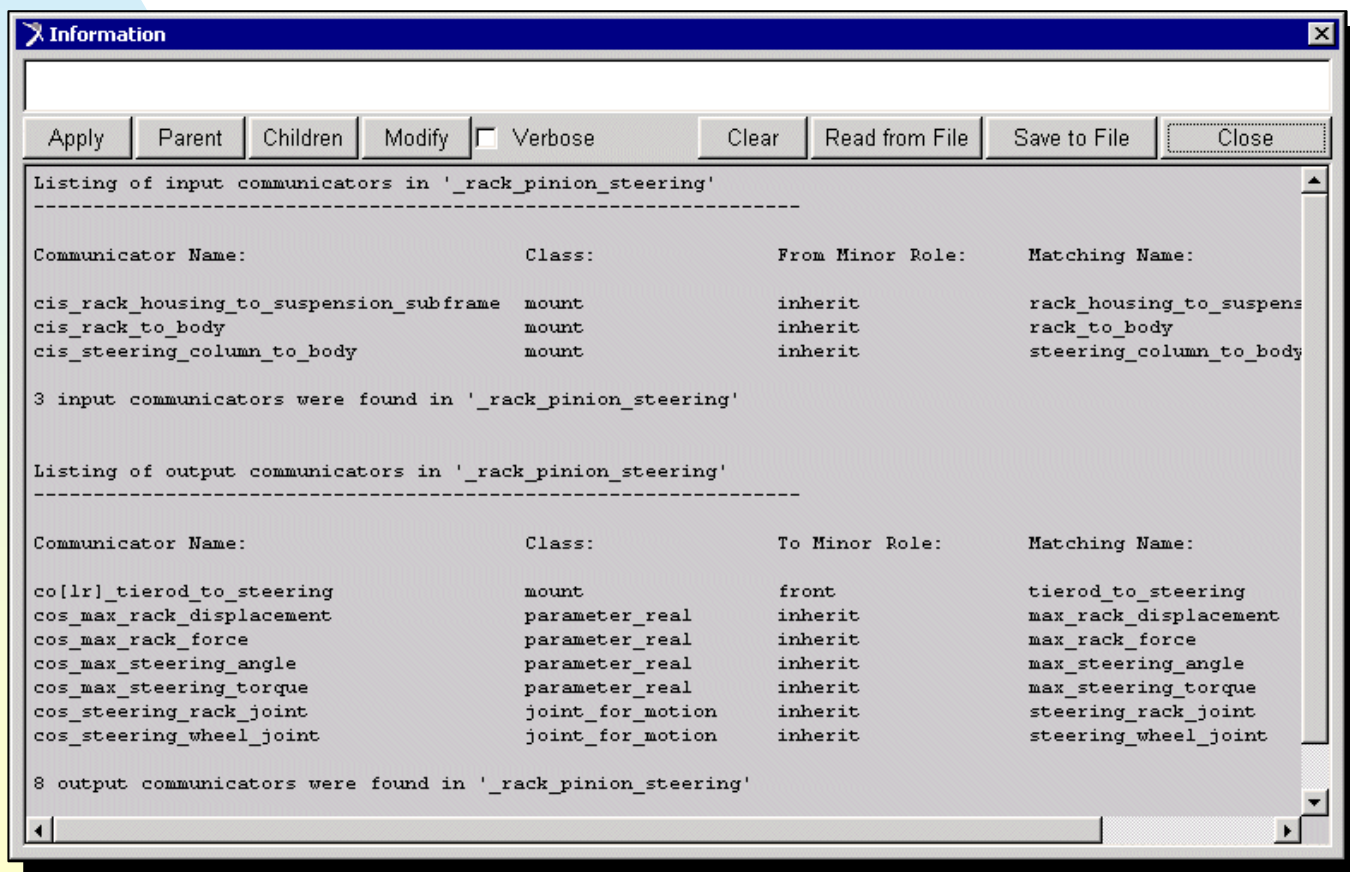
*注意：*模型的名字文本栏内自动地显示当前激活的模板的名字。

■ 要列出你的模板中所有的信息交换器的信息：

1. 在 **Communicators Info** 对话框内设置 **Entity** 为 **All**。
2. 选择 **OK**。

练习 10 取得信息交换器的信息

在信息窗口内列出了在齿轮齿条转向模板内所有的信息交换器的信息如下图所示：



练习 10 取得信息交换器的信息

■ 检查信息交换器

- ◆ 在本节，你将要设置工作空间，这样你可以看到你的齿轮齿条转向模板中的信息交换器与悬架模板中信息交换器是否相匹配。你将打开一个悬架模板，通过取得信息交换器的信息，你将可以确定必须在模板中作些什么才能保证装配正确进行 (例如，你必须建立哪些信息交换器)。

注意：确认所有你想测试的模板都在 ADAMS/Car 的进程中被打开。

练习 10 取得信息交换器的信息

■ 要打开一个模板：

1. 从菜单 **File** 下选择 **Open**。
2. 在 **Template Name** 文本栏内指向 **Search**，并选择 **<acar_shared>\templates.tbl**。
3. 双击 **_double_wishbone_torsion.tpl**。
4. 选择 **OK**。

一个与齿轮齿条转向模板没有关系的模板被打开了，原来的模板仍然处于打开状态。现在显示在你的工作空间的是 double-wishbone suspension 模板。

■ 要执行一个信息交换器的测试：

1. 从菜单 **Build** 下指向 **Communicator** 再选择 **Test**。
2. 清除 **Model Names** 文本栏内的内容。
3. 在 **Model Names** 文本栏内指向 **Model** 再指向 **Guesses**，然后再选择 **_double_wishbone_torsion**。
4. 重复步骤 3 以选择 **_rack_pinion_steering**。

练习 10 取得信息交换器的信息

5. 在 **Minor Roles** 文本栏内输入信息交换器的次要角色 (例如, front)。你需要对你所选择的每个模板或试验台都分别输入一个次要角色。
 - ◆ **注意：**每个信息交换器都有一个次要角色，缺省的，为any、front、rear、trailer 或 inherit中的一个。inherit 次要角色指定当 ADAMS/Car 从该模板生成子系统时，该模板应该继承子系统的次要角色。因为在你测试一个模板的信息交换器时，其继承的次要角色尚未定义，在 Minor Role 文本区域输入次要角色可以提供信息交换器的次要角色。例如，如果你分配模板 susp_02 的次要角色为 front，那么信息交换器测试中将改变 susp_02 下所有的信息交换器的次要角色为继承模板的次要角色，即为 front。
6. 如果你此前曾经使用过信息窗口，选择 **Clear Information Window**。

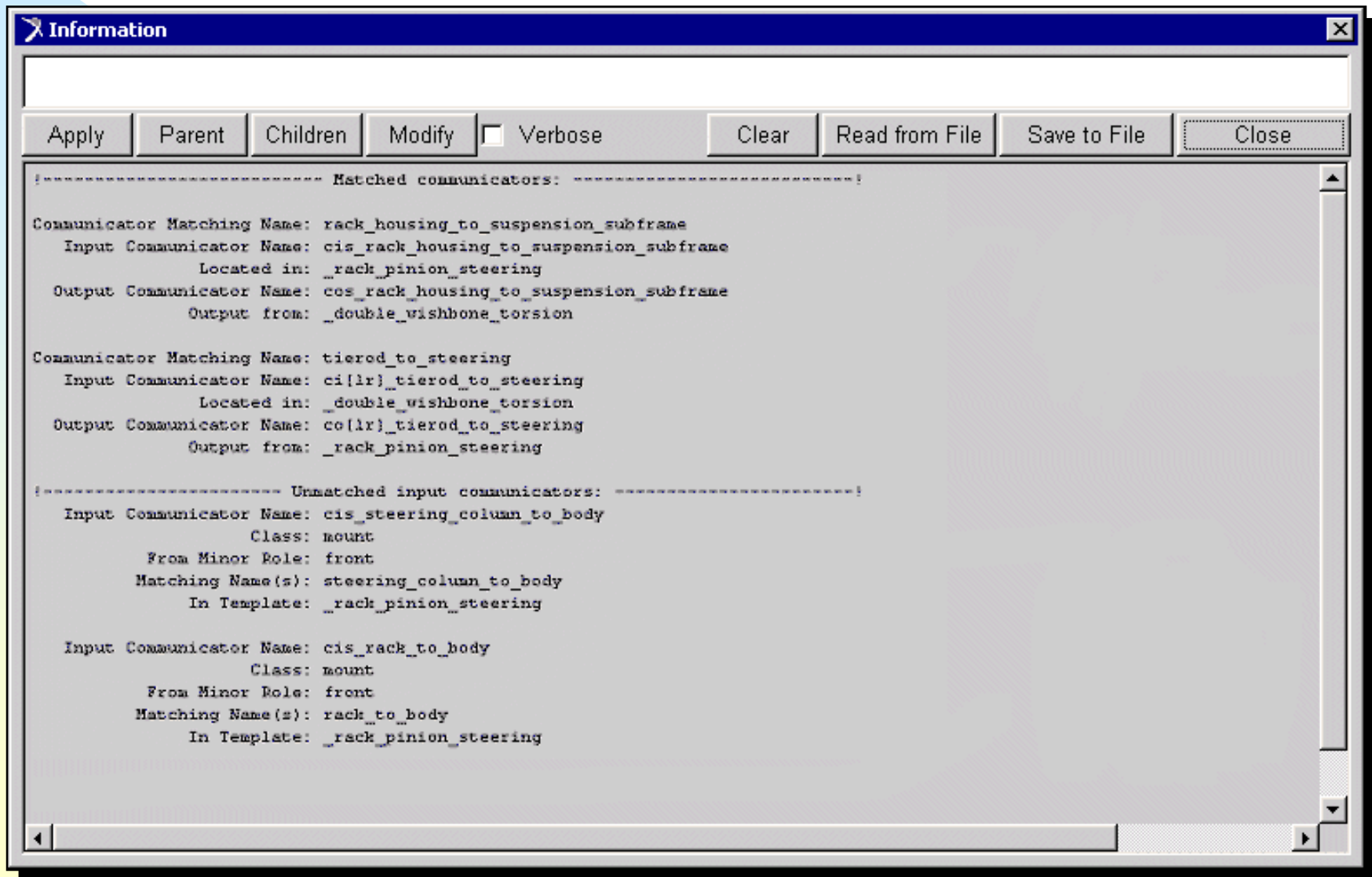
练习 10 取得信息交换器的信息

7. 如果你存储仿真结果，选择 **Save Test Results to File**。
8. 在 **File Name** 文本栏内，输入文件名称。
9. 选择 **OK**。

窗口 Information，如下页所示，列出了所有的信息交换器，包括与其它信息交换器匹配的和不对应的。先显示匹配的，后面显示不对应的。列表其中包括输入输出信息交换器以及其所属的模板的名称。通常情况，你将看到很多不对应的信息交换器，这些信息交换器通常是与你在当前工作空间未曾打开的模板或试验台相关的。

在本例中，注意信息交换器 rack_housing_to_suspension_subframe 和 tierod_to_steering 是匹配的。然而，信息交换器，如 rack_to_body 和 max_rack_force 则是不匹配的，因为，需要其信息的模板没有选上或没有打开。

练习 10 取得信息交换器的信息



练习 10 取得信息交换器的信息

- ◆ 要逐步了解在一个模板里信息交换器的拓扑关系，研究输入和输出，并看看其与其它模板的关系。当你在一个新的模板中生成信息交换器时，一个好的方法是打开一个现有的模板，并据此生成相同的信息交换器。
- ◆ 最后要提示的是，一定要注意输出信息交换器只是发布一些信息，因此，它们只是被动的，不会影响子系统的行为。但是，输入信息交换器则需要搜索其所需要的信息，它们将会影响你的装配和仿真，尤其在它们不匹配的情况下。

练习 10 取得信息交换器的信息

- 有关装配和位置信息交换器更多信息，参见下列KBAs：
 - ◆ Article 9182 -- Use of mount and location communicators at:
<http://support.adams.com/kb/faq.asp?ID=9182>
 - ◆ Article 9184 -- Example of use of location communicators at:
<http://support.adams.com/kb/faq.asp?ID=9184>
 - ◆ Article 9185 -- Example of use of mount communicators at:
<http://support.adams.com/kb/faq.asp?ID=9185>

第 12 章 使用弹性体

- 在本章中，你将学习如何在你的模型中考虑弹性部件，以及如何由刚性部件替换为弹性部件。

使用弹性体

- 本章内容：
 - ◆ 弹性部件综述
 - ◆ 使用弹性部件的一些限制
 - ◆ 得到弹性部件
 - ◆ 模态叠加
 - ◆ 可视化属性
 - ◆ 有关约束和驱动
 - ◆ 约束连接的限制

弹性部件综述

- ADAMS/Flex 使用模态叠加的方法来模拟弹性部件，称作模态综合。模态综合在弹性体上分配一系列的模态振型。这种模拟弹性的模态综合法在模拟具有高弹性及适度的变形上非常有用，具体来讲，变形大小在该部件的特征长度上小于 10% 的情况下适用。

弹性部件综述

- 通过在你的模型中集成弹性部件，你可以实现：
 - ◆ 捕捉操纵性和舒适性仿真过程中部件的惯性和柔顺性特性。
 - ◆ 允许MSC.ADAMS 在仿真过程中可以考虑部件的弹性，从而得到更高精度的预测载荷。
 - ◆ 研究变形 (结合 ADAMS/Durability ，还可以计算应力/应变)。
 - ◆ 在使用 ADAMS/Flex 时，结合 ADAMS/Linear ，可以检查一个弹性模型的线性化系统模态。
- 当你预计部件的弹性将影响你的模型的动态性能或你需要你的模型中一个部件变形的精确信息时，你应该使用弹性体。

使用弹性部件的一些限制

- 在你使用弹性体时，一定要注意弹性体的变形一系列变形的线型叠加的结果。因此，在模拟高阶变形时要特别小心，比如出现大的变形或要试图准确的模拟一个转子系统的离心刚度。你可以将弹性体分解成若干个子结构再在 ADAMS/Car 中将它们组装起来的办法来克服这些限制。
- 同样的，注意弹性体是没有参数化的。如果你想在你的系统中替换一个新的弹性体，你必须重新生成新的弹性体模型。

得到弹性部件

■ 要生成弹性体，有两种方式：

- ◆ 导入模态中性文件 (.mnf) – 要生成一个新的弹性体，可以从 Build → Part → Flexible Body。ADAMS/Car 将导入一个 .mnf 文件并生成弹性体。
- ◆ 使用 ADAMS/AutoFlex 生成 .mnf 文件 – 使用附加模块 (需要另外的 license)，一个弹性体可以不需要借助外部的 FEA 软件就可以生成。指定截面形状、中心线和连接点。此工具生成弹性体，与导入 .mnf 一样。在 ADAMS/Car 中，ADAMS/AutoFlex 将还允许你生成卡车车架 (two rails with multiple cross-members)。



- 有关 ADAMS/AutoFlex 更多信息，参见 ADAMS/AutoFlex 在线帮助。

模态叠加

- ◆ ADAMS/Flex 处理弹性体的变形为在大位移运动基础上相对于部件上一个本地参考坐标系小的线性变形。
- ◆ 变形表示为模态振型的线性叠加

$$\mathbf{u} = \sum_{i=1}^M \phi_i q_i$$

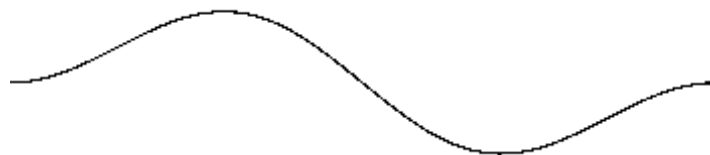
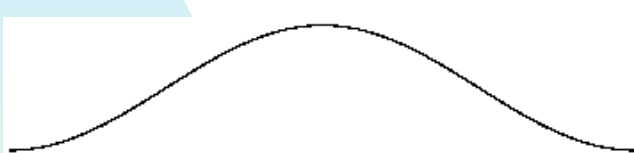
- ◆ 示例



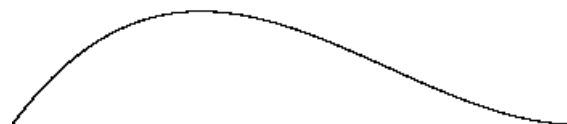
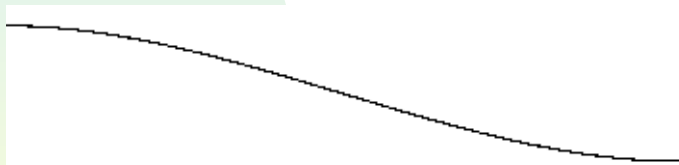
- ◆ Craig-Bampton 模态将外部节点与内部节点相区别开来。

模态叠加

- ◆ 固定边界节点的正交模态



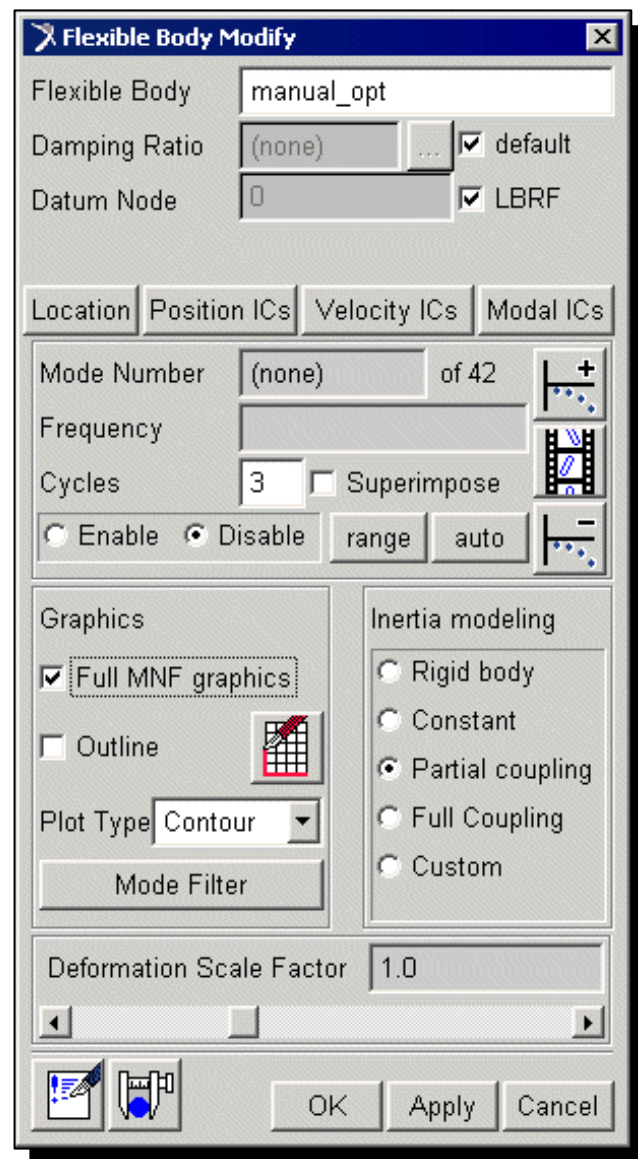
- ◆ 约束模态



欲了解更多 ADAMS/Flex 的理论背景，参见 ADAMS/Flex 的在线帮助 (特别是，在转换 FE 的模型时，参见 Craig-Bampton 模态部分)。

可视化属性

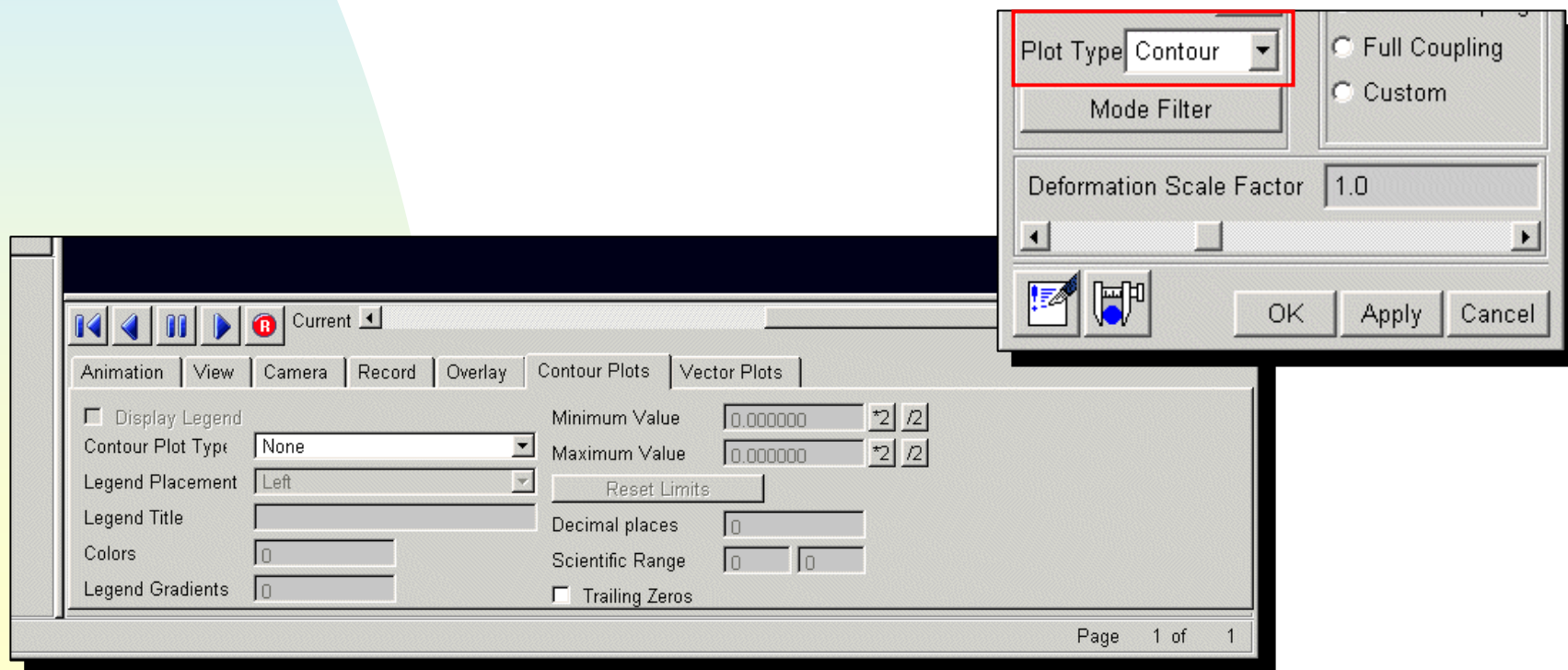
- 你可以使用 Plot Type选项以设置或者云图或矢量图的显示。



可视化属性

■ 云图颜色

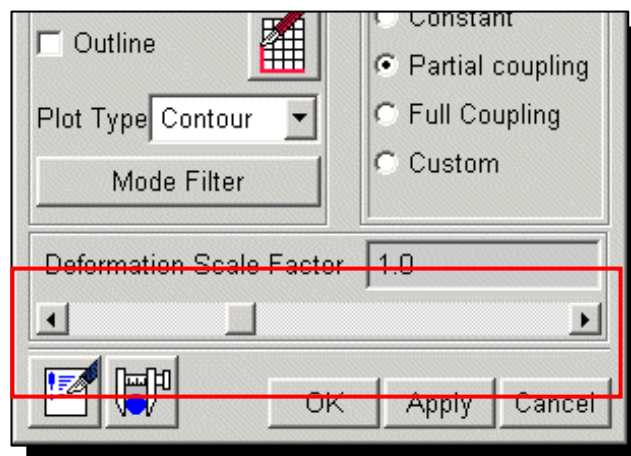
- ◆ 表示弹性部件变形的大小
- ◆ 为连续的，不是离散的节点值
- ◆ 显示相对的变形，而不是应力
- ◆ 在 ADAMS/PostProcessor 中云图的开关由 Contour Plots 按钮控制。



可视化属性

■ 变形比例因子

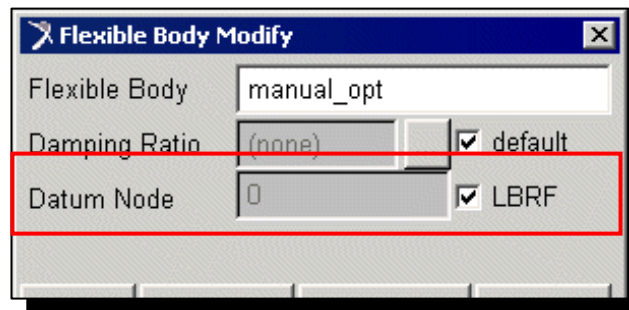
- ◆ 用于放大变形
- ◆ 可以放大或缩小
- ◆ 在变形比例因子 > 1 的情况下，约束可能会出现不一致的情形。这只是视觉上的效果，但分析当然是保证你所定义的约束保持一致。
- ◆ 当变形比例因子等于 0 时，不显示变形，但显示颜色。



可视化属性

■ 数据节点

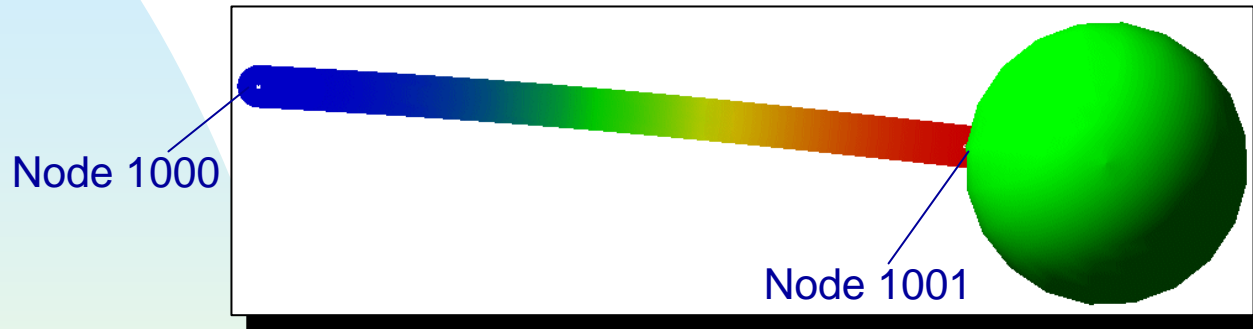
- ◆ 变形是一个相对量，可以表示为相对于一个节点的 量：
 - 你选择一个节点，ADAMS/Flex 将其定义为变形的基准节点，其它节点的变形为相对于该点的变形量
 - 节点的变形采用相对于该节点的颜色代表
 - LBRF (局部部件参考坐标系，local body reference frame，缺省的，即是部件坐标系或 BCS)，与在有限元软件中的参考坐标系的位置相同。



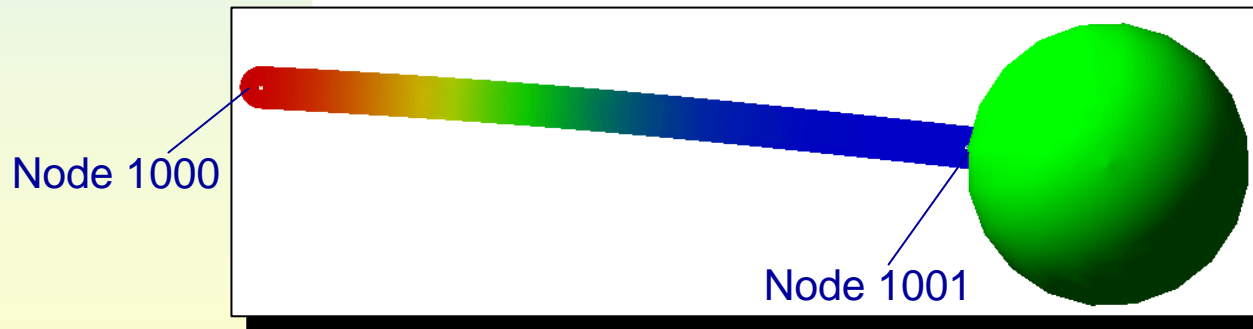
可视化属性

■ 使用不同变形基准节点的例子

- ◆ 此处，节点1000位变形基准节点



- ◆ 此处，节点1001位变形基准节点



- ◆ 红颜色代表相对于变形基准节点最大变形位置。

有关约束和驱动

■ 使用约束

- ◆ 在 ADAMS/Car 中引入弹性体后，你就可以使用 ADAMS/Car 中所提供的约束库将其与模型中的其它刚性部件相连了。

■ 约束的位置

- ◆ 直接连接到弹性体上的约束的位置必须是在节点上。
- ◆ 连接的节点并不一定是有限元中的外接点。但从较好的建模经验来看，最好是在外接节点上进行约束连接。

■ 你需要避免节点不匹配：

- ◆ 使用一致的编号规则
- ◆ 特别注意节点的排列问题

约束连接的限制

- 用于连接弹性体的常用约束：

- ◆ Fixed
- ◆ Revolute
- ◆ Spherical
- ◆ Universal (or Hooke)

- 下面的约束不能直接连接弹性体：

- ◆ 有运动驱动的约束
- ◆ 任何允许移动的约束 (translational, planar, 等等)
- ◆ 任何允许移动的基本约束 (inline, inplane, 等等)

解决办法：连接该约束到中间的哑物体上 (例如，一个中间部件 **interface part**)，该哑物体与弹性体在节点上以固定副相连。

练习 11 弹性部件入门练习

- 在本练习中，你将在一个悬架模板中的刚性下拉臂替换为一个弹性的下拉臂。
- 首先，在 ADAMS/Car 模板建模状态下，打开 double-wishbone suspension template。



本练习大约需要一个半小时完成。

练习 11 弹性部件入门练习

■ 打开一个子系统

要打开一个子系统:

1. 从菜单 **File** 选择 **Open**。
2. 在子系统 **Name** 文本窗口内电击鼠标右键，再指向 **Search**，然后选择 **<acar_shared>\subsystems.tbl**。
3. 双击 **TR_Front_suspension.sub**。
4. 选择 **OK**。

练习 11 弹性部件入门练习

■ 生成一个弹性体

- ◆ 在刚体上生成弹性体，并将刚体置换为弹性体。当没有任何对象参照刚体时，你可以将刚体删除。

■ 要生成弹性的下拉臂：

1. 从菜单 Adjust 选择 General Parts 再选择 Rigid-to-Flex。
2. 在 Current Part Name 文本栏内输入 gel_lower_control_arm。
3. 在 **Left** 和 **Right Modal Neutral File** 文本栏内，选择在数据库 <acar_shared> 中的 **LCA_left_shl.mnf** 和 **LCA_right_shl.mnf** 两个文件。

ADAMS/Car 在刚体上直接显示弹性体。当模型处于线图显示时，你能够看见该物体。

提示：如果你的模型不是处于线图显示时，你可以按大写的 S，将模型显示设置为线图方式。



练习 11 弹性部件入门练习

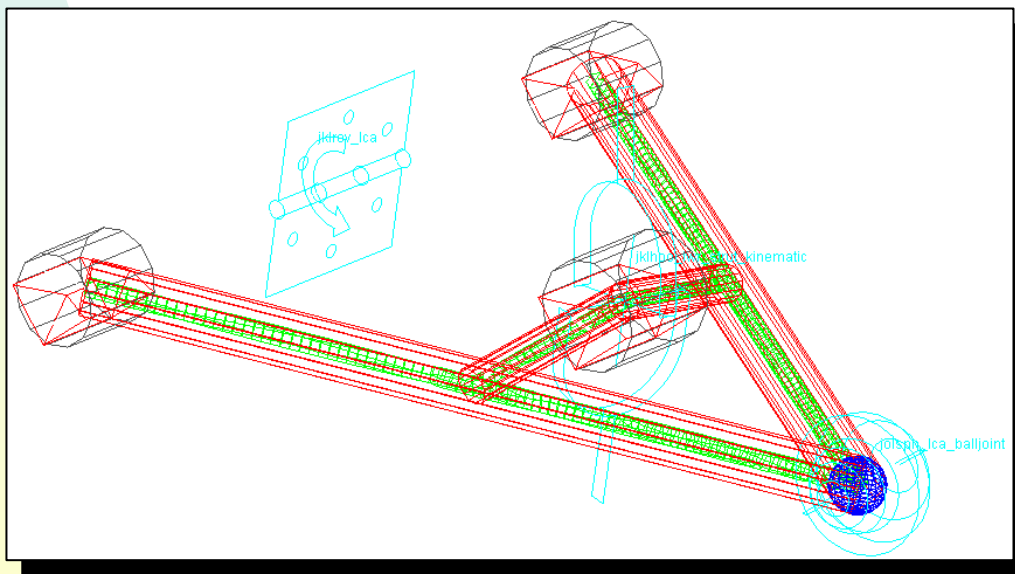
■ 识别外接点

- ◆ 现在你需要为弹性体在每个外接点处生成连接部件，在这些外接点处，你的模型都有与下拉臂的连接关系。然后，你替换相应所生成的连接部件。
- ◆ 在该模型中， `gel_lower_control_arm` (刚性的下拉臂)有六处连接关系，如下页图所示。你需要识别这些连接关系，以替换连接部件。注意，某些连接关系是否起作用，取决于模型是否为运动学分析模式。

练习 11 弹性部件入门练习

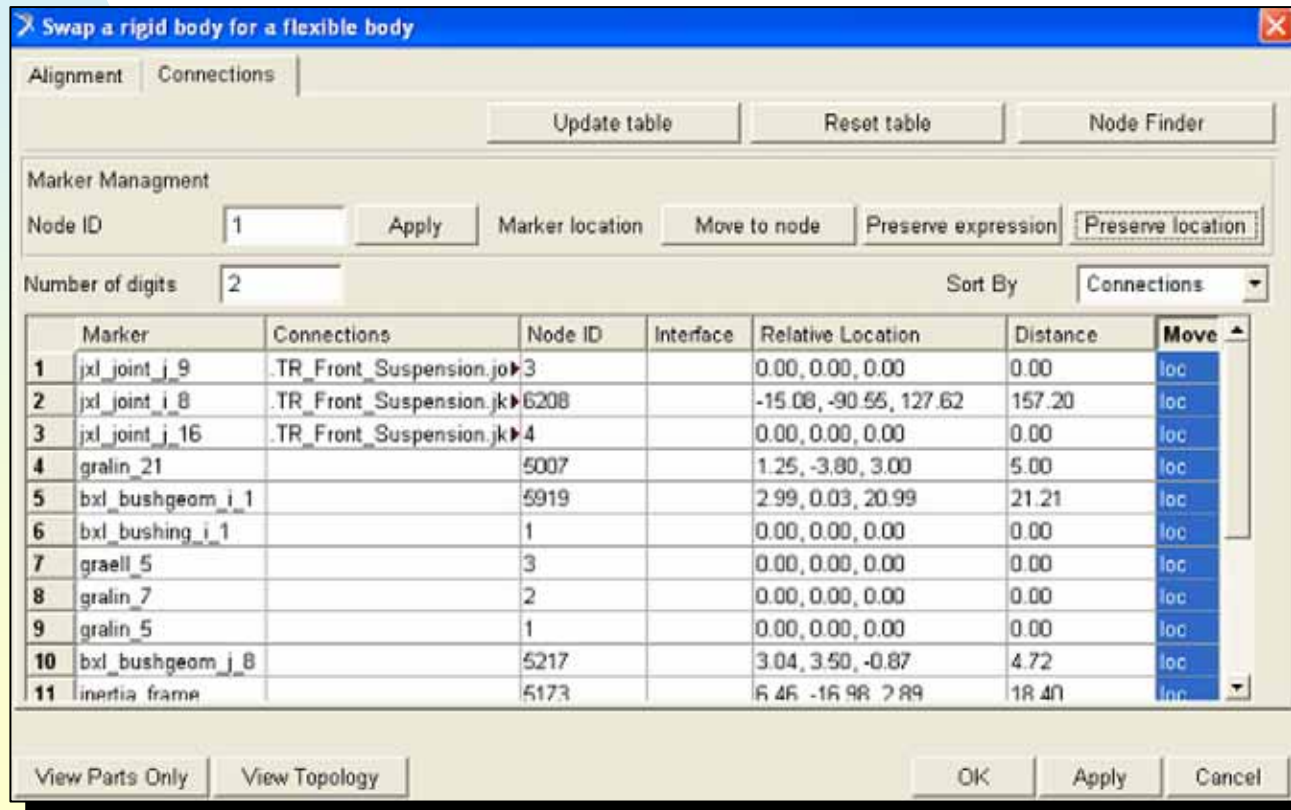
要识别外接点：

1. 在 **Status** 栏内，右键点击 **Information** 工具包 ，再选取 **Model Topology by Parts** 工具 。
在显示的信息窗口内出现模型的拓扑关系。
2. 滚动到 **Information** 窗口的顶部并寻找 **gel_lower_control_arm**，找到相关的连接关系，你需要修改这些连接以使用弹性体，而不是刚体。



练习 11 弹性部件入门练习

- 使用 Preserve location 按钮，重置连接弹性体的连接关系。
 1. 选择 Move 按钮，亮显所有的连接关系。
 2. 下一步选择 Preservers locations。



练习 11 弹性部件入门练习

1. 当你改变了连接关系后，试着删除刚性下拉臂。
2. 如果刚体下拉臂仍然有其它的连接关系，ADAMS/Car 会提示一个警告信息，并指示你进行下面的操作：
 - **Continue** - ADAMS/Car 删除你所选择的对象以及其它与你所选择的对象有关联的对象。
 - **Highlight & List Dependents** - ADAMS/Car 列出模型中存在的所有相关的依存关系，并且高亮显示这些对象。
 - **Cancel** - ADAMS/Car 不作任何修改。如果你选择 Cancel，你必须在手工删除所有的相关对象之后，ADAMS/Car 才能删除该对象。

练习 11 弹性部件入门练习

- 显示的警告信息表明：弹性体不能再通过简单的移动模型中硬点的位置而改变形状了。



练习 11 弹性部件入门练习

■ 要删除悬架参数组：

1. 悬架参数组是定义用来计算悬架输出的转向轴的，要删除悬架参数组，可以从菜单 **Build** 下选择 **Suspension Parameters**，再选择 **Characteristic Array**，然后再选择 **Delete**。

悬架参数组包括原来的刚体部件：gel_lower_control_arm在内，你需要使用合适的连接部件来替代。但是，悬架参数组只能对应于一种配置，因此，你需要先将其删除，然后，再重新按照新的连接部件设置。

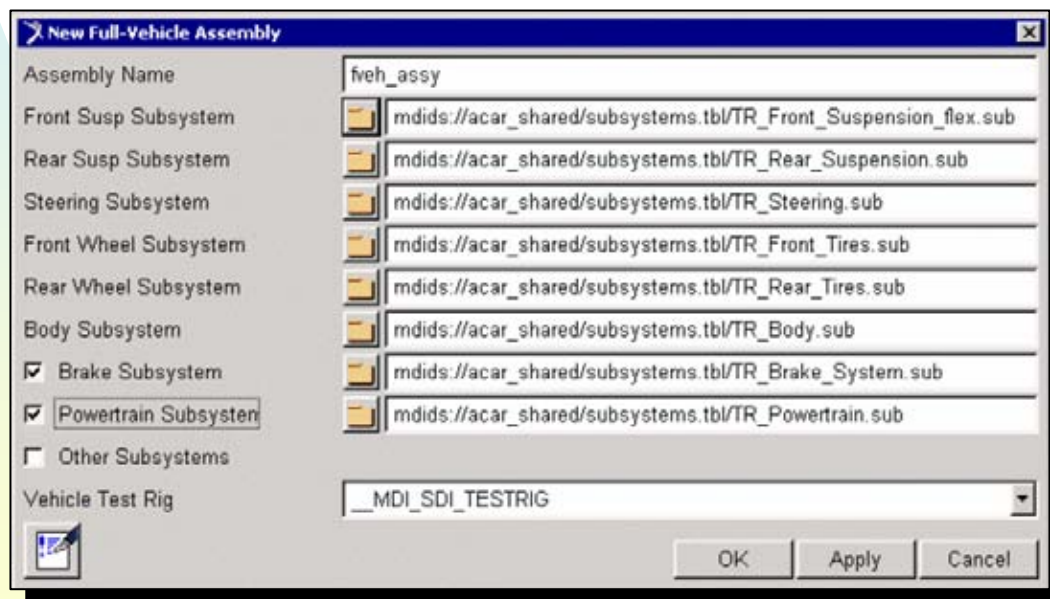
练习 11 弹性部件入门练习

2. 从菜单 **Build** 下选择 **Suspension Parameters** 再选择 **Characteristic Array** , 再选择 **Set** , 并对悬架参数组作如下的改变 (几何上) :
 - **I Part:** 在 hpl_lca_outer 处的连接部件 (其名字取决于你如何赋予)
 - **J Part:** gel_upper_control_arm
 - I Coordinate Reference:** hpl_lca_outer
 - J Coordinate Reference:** hpl_uca_outer
- ◆ 重复上述过程, 完成右侧的修改(替换 ger_lower_control_arm 为 LCA_right_shl.mnf)。

练习 11 弹性部件入门练习

■ 要对包括弹性下拉臂的模型进行分析：

1. 在数据库 acar_training 中存储子系统 TR_Front_Suspension_flex。
2. 生成装配
 - 从菜单 **File** 下选择 **New** 再选择 **Full-Vehicle Assembly**。
 - 按照下图填充对话框，再选择 **OK**。



练习 11 弹性部件入门练习

■ 定义运动学分析模式

- ◆ 要减少连接陈套对子系统柔顺性的影响，你需要设置悬架子系统的分析模式为运动学模式，这样的话，只考虑弹性部件下拉臂的弹性。

■ 要定义运动学分析模式：

1. 从菜单 View 下选择 Subsystem。
对话框 Display Subsystem 显示，默认的子系统为：
fveh_assy.TR_Front_Suspension_flex。
2. 选择 OK。
ADAMS/Car 显示悬架子系统。
3. 从菜单 Adjust 下选择 Kinematic Toggle。
4. 设置 Current Mode 为 Kinematic。
5. 选择 OK。

练习 11 弹性部件入门练习

■ 设置模态

- ◆ 缺省情况下，当你集成一个 MNF 到 ADAMS/Car 的模板时，ADAMS/Flex 会将有限元模型中生成的除了刚体模态之外的所有模态设置为激活状态。适当地设置弹性体的模态非常重要，因为一个 MNF 文件中所包含的模态比实际上进行仿真以得到特定的响应所需要的模态要多。
- ◆ 为了提高仿真的效率，你应该将弹性体上某些与系统运动过程关系不大或者说影响很小的模态失效掉。在失效某些模态时需要十分小心，因为失效掉某些模态对应于在部件上增加了一些约束。改变一个弹性体的模态即为弹性体的设置过程。

练习 11 弹性部件入门练习

- ◆ 你可以手工切换模态的“on”或者“off”。这可以给你极大的弹性，有助于避免一些可能存在的问题。ADAMS中提供下面几种激活或失效模态的方式：
 - 基于其模态阶数，对每阶模态分别处理
 - 基于其模态阶数或频率，对一组模态进行处理
 - 通过表编辑的方式。表编辑方式同时允许你设置与每阶模态相关的位移和速度的初始条件。
 - 基于其对应变能量的贡献大小进行设置。但采用这种方式，你需要在成功的作一次分析之后才能进行。有关这种方式的更多信息，可参见ADAMS/Flex 的在线帮助手册。

练习 11 弹性部件入门练习

■ 要失效某些模态：

- ◆ 弹性下拉臂的某些模态对于整个系统的动态性能影响很小。建议 你将这些模态失效掉，以减小计算量，这样可以提高计算的效率。
- ◆ 你需要对左右两侧的弹性下拉臂的这些模态同时失效。
 1. 先框选前悬架。
 2. 从菜单 **Adjust** 选择 **Flexible Body** 再选择 **Mode Manager**。
出现 Flexible Body Modify 的对话框。

练习 11 弹性部件入门练习

3. 在 **Flexible Body** 文本栏内点击鼠标右键，再选择 **Flexible Body**，选择 **Pick**，然后从屏幕上选择左侧的下拉臂 **fbl_lower_control_arm**。
4. 选择 **Modal ICs**。
出现 **Modify Modal ICs** 对话框。
5. 按住 **Shift** 键，选择模态 **28** 和 **29**，再选择 **Disable Highlighted Modes**。
6. 关闭 **Modify Modal ICs** 对话框。
7. 重复步骤 3 到 6，完成右侧的下拉臂 **fbr_lower_control_arm** 的设置。
8. 关闭 **Flexible Body Modify**对话框。

练习 11 弹性部件入门练习

■ 完成整车分析

- ◆ 现在，你可以进行整车的分析了。分析完成后，你可以改变弹性体的惯性模式，以比较部件弹性对车辆动力学性能的影响。

■ 要完成整车分析：

1. 从菜单 **Simulate** 选择 **Full-Vehicle Analysis**，再选择 **Open-Loop Steering Events**，再选择 **Step Steer**。
2. 按照下面选项设置分析：
 - **Full-Vehicle Assembly:** fveh_assy
 - **Output Prefix:** t st
 - **End Time:** 4

练习 11 弹性部件入门练习

- **Number of Steps:** 400
 - **Initial Velocity:** 70 (take the default of km/hr)
 - **Gear Position:** 3
 - **Initial Steer Value:** 0
 - **Final Steer Value:** -45
 - **Step Start Time:** 1
 - **Duration of Step:** 1
 - **Steering Input:** Angle
3. 保持 Cruise Control (off) 和 Quasi-Static Straight Line Setup (on)按照缺省设置不变。
4. 选择 **Apply** , 这样 , 对话框保持打开状态 , 你将在后面的章节运行分析。
- SDI 试验台施加你所指定的输入到装配上并完成动力学分析。

练习 11 弹性部件入门练习

■ 要改变惯性模型：

1. 双击左侧下拉臂 **fbl_lower_control_arm**。
2. 选择 **Mode Manager** 工具。
3. 设置 **Inertia modeling** 为 **Rigid body**。
4. 选择 **OK**。
5. 关闭 **Modify Flexible Body** 对话框。
6. 使用同样的输入提交另一次阶跃转向分析，但是将输出的钱缀设为 **tst_rigid**。

ADAMS/Car 进行该装配的分析。

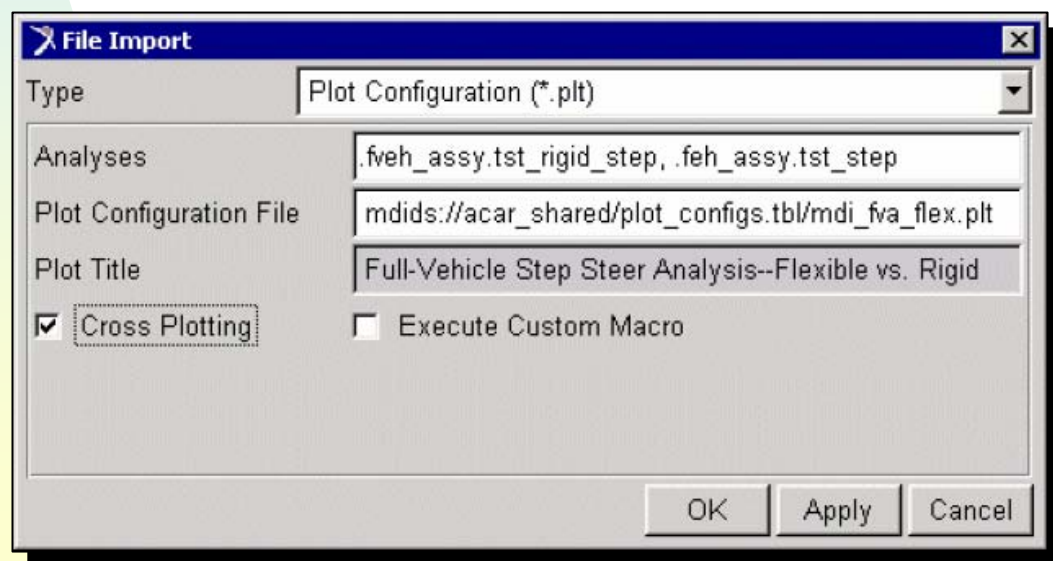
练习 11 弹性部件入门练习

■ 绘制分析结果曲线

- ◆ 在这一节中，你将生成一系列的曲线，表示你的车辆系统的性能，同时，观察一下下拉臂的弹性对车辆动力学性能的影响。

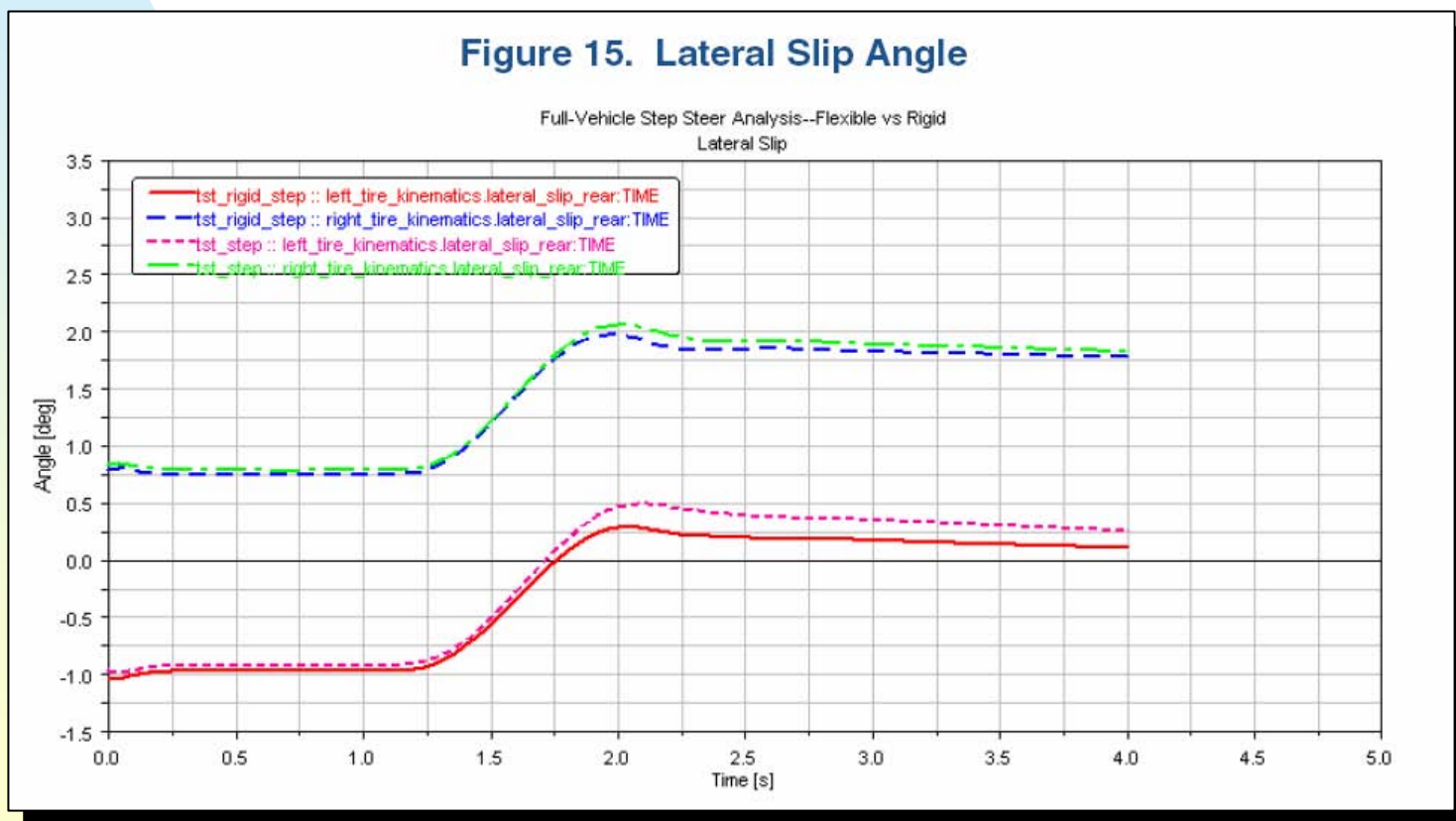
■ 要绘制分析结果曲线：

1. 启动 ADAMS/PostProcessor.
2. 从菜单 **Plot** 选择 **Create Plots**。
3. 按照下图填写对话框，再选择 **OK**。



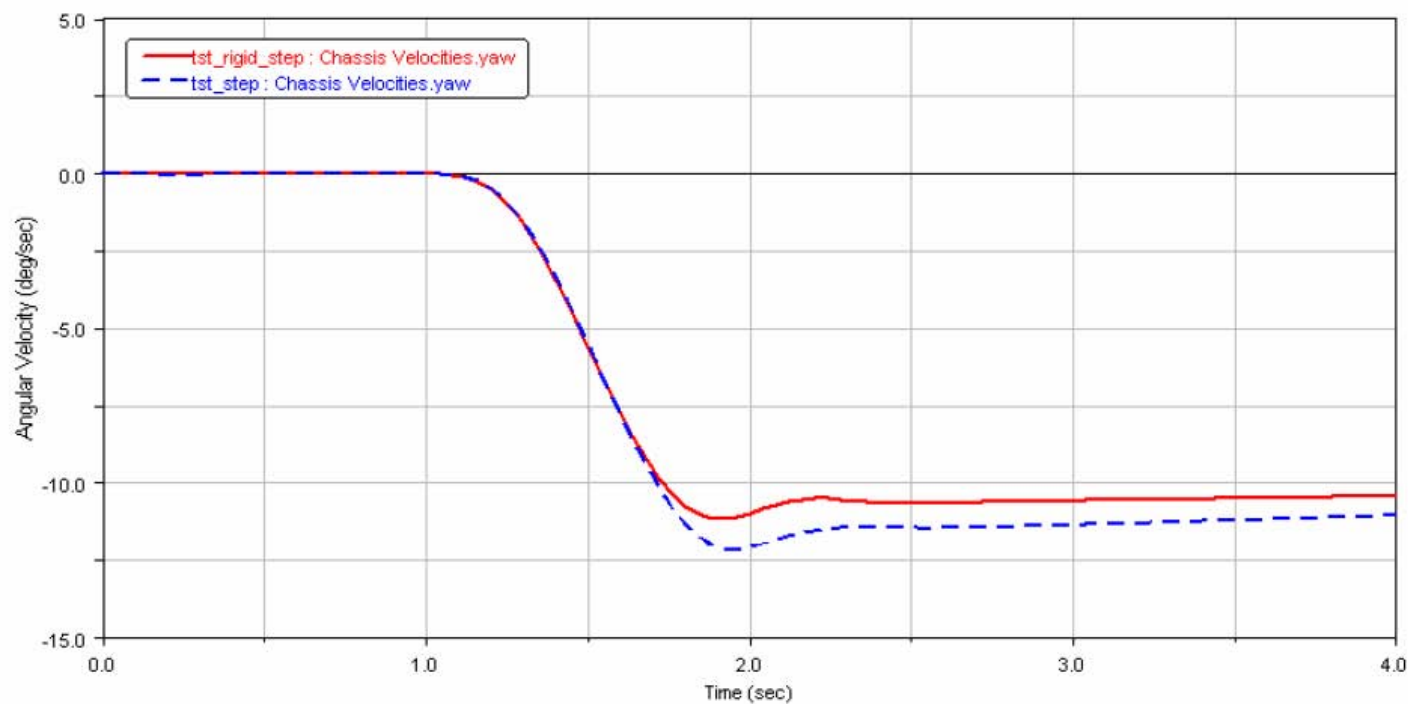
练习 11 弹性部件入门练习

- ◆ 图 15 和 16 显示一些曲线。从图中可以看到下拉臂的弹性的影响。
- 4. 观察一下曲线再返回 ADAMS/Car。



练习 11 弹性部件入门练习

Figure 16. Yaw Rate



第 13 章 测试请求

- 本章介绍测试请求（requests），测试请求为 ADAMS/Car 中最主要的输出数据方式。

测试请求

- 本章内容：
 - ◆ 生成新的测试请求
 - ◆ 测试请求的类型
 - ◆ 测试请求激活的切换

测试请求激活的切换

- 在子系统或装配中可以切换测试请求的激活性。任何激活性的改变都将只影响当前进程，因为此项设置并不存储在子系统文件 (.sub) 中。
- 可以进行激活/失效操作的测试请求包括：
 - ◆ actuators
 - ◆ bushings
 - ◆ springs
 - ◆ dampers
 - ◆ bumpstops
 - ◆ reboundstops
 - ◆ 所有的 (上述类型)
- 要存储测试请求的激活性，你可以建立一个包含参数变量的组，该变量是存储在子系统文件中的。

测试请求激活的切换

■ 要在模板建模状态下存储测试请求的激活性：

1. 生成一个存储测试请求激活性的参数化变量，1 为激活的，0 为失效的。例如：pvs_request_activity.
2. 生成一个组，使用命令：**Tools → Command Navigator, group → create**



更多有关组的信息，按 F1 键。

3. 在组中添加你的测试请求。

测试请求激活的切换

4. 对于该组的参数 **expr_active** (1 = active, 0 = inactive), 生成一个函数, 使用前面步骤 1 中所生成的变量。

此项操作的命令类似于下面的命令：

```
group create &  
  group_name = ._my_template.my_request_activity  
  objects_in_group = ._my_template.request0, &  
    ._my_template.request1, &  
    ._my_template.request2, &  
    ._my_template.request3, &  
  expr_active = (pvs_request_activity)
```

更多的例子, 可以参考在共享数据库中的模板, 其中包括一个名为 kinematic_mode_active 的组, 用来设置运动学/动力学分析模式之间的切换。

第 14 章 轮胎

- 本章中介绍轮胎力的计算以及可以使用的轮胎模型。

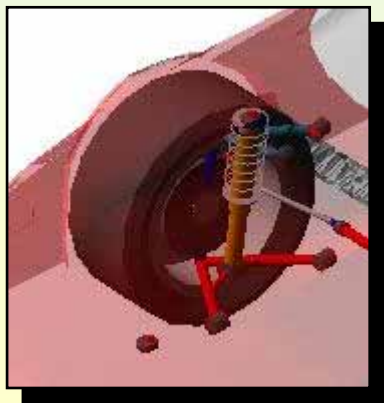
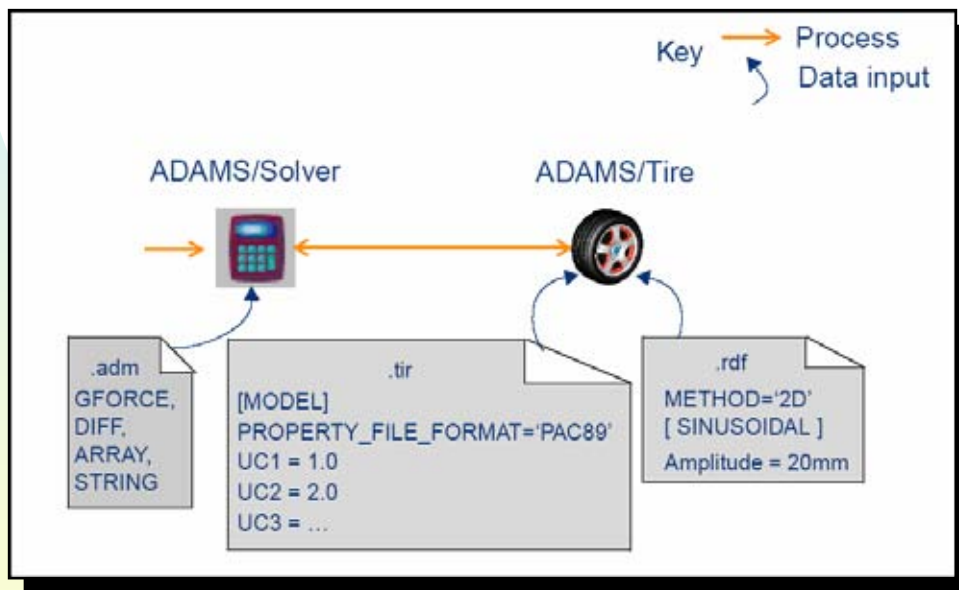
TIRES

■ 本章内容：

- ◆ 轮胎综述
- ◆ ADAMS/Tire 模块
- ◆ Tire 模型
- ◆ 路面类型
- ◆ 2D 路面的例子 (RAMP)
- ◆ 使用 3D 等效体积模型
- ◆ 3D 路面的例子
- ◆ 3D 光滑路面
- ◆ 路面数据文件的实例
- ◆ 使用 3D 光滑路面
- ◆ 如何使用 ADAMS/Tire

轮胎综述

- ADAMS/Tire 用来计算作用在车辆上轮胎所受到的力和力矩，该力和力矩是基于轮胎和路面之间相互作用情况所得到的。



轮胎综述

- ADAMS/Tire 为 ADAMS/Solver 在解算过程中调用的一系列共享目标文件库，包括 DIFSUB 和 GFOSUB 子程序。
- 你能够在车辆的操稳性能或耐久性分析中使用 ADAMS/Tire 来模拟轮胎：
 - ◆ 操纵稳定性分析用于研究车辆在得到如转向、制动或油门等输入信号下的动力学响应。例如：你可以分析当车辆在以某一个速度行驶时给方向盘一个转角情况下，其横向加速度的变化情况。
 - ◆ 耐久性分析用于产生路面载荷历程以及进行应力强度、疲劳分析所需要的部件载荷及加速度的计算。此项研究有助于你计算各种路面形状如：坑洼路面、路边台阶或比利时路面等的影响。
- ADAMS/Tire 允许你在一个车辆模型中最多可以有 40 个轮胎。

ADAMS/Tire 模块

- ADAMS/Tire 有一系列的轮胎模块，你可以结合 ADAMS/View、ADAMS/Solver、ADAMS/Car和 ADAMS/Chassis使用。这些模块使你能够模拟常见的各种车辆如：轿车、卡车或飞机上的橡胶轮胎。特别是，这些轮胎模块可以模拟轮胎上产生的力以使车辆加速、减速或转向等。在 ADAMS/Tire 中可以用的轮胎模型有：
 - ◆ ADAMS/Tire Handling Module
 - ◆ ADAMS/Tire 3D Contact Module
 - ◆ ADAMS/Tire 3D Road Module
 - ◆ Specific Tire Models
 - ◆ Features in ADAMS/Tire Modules

ADAMS/Tire 模块

■ ADAMS/Tire 操纵性轮胎模块

- ◆ ADAMS/Tire 操纵性轮胎模块包含下列可用于车辆动力学研究中的轮胎模块：
 - Pacejka 2002 tire model*
 - Pacejka '89 and Pacejka '94 models*
 - Fiala tire model
 - UA tire model
 - 5.2.1 tire model
- ◆ ADAMS/Tire操纵性轮胎模块采用 point-follower 的方法计算轮胎的正压力并且限于二维的路面模型。

注释：* Pacejka 轮胎模型中所使用计算公式来源于 H.B. Pacejka 博士公开发表的文献，在汽车行业通常指的是 Pacejka 模型。Dr. Pacejka 本人既没有参与这些轮胎模型的开发，也未以任何方式资助其开发。

ADAMS/Tire 模块

■ ADAMS/Tire 3D Contact 模块

- ◆ ADAMS/Tire 3D Contact 模块采用三维等效路面方法计算在三维路面上的轮胎正压力，用来预测车辆上所受到的载荷以进行耐久性能研究。当你同时购买了 ADAMS/Tire 3D Contact 模块和ADAMS/Tire 操纵性轮胎模块，你可以使用 Pacejka 2002、Pacejka '89、Pacejka '94 或 Fiala 轮胎模型以计算轮胎的操纵力和力矩 (横向力、纵向力、矫正力矩等等)。如果你只购买了 ADAMS/Tire 3D Contact 模块，你就只能使用 Fiala 轮胎模型以计算轮胎的操纵力和力矩。

ADAMS/Tire 模块

■ ADAMS/Tire 3D 路面模型

- ◆ ADAMS/3D路面模型允许你定义任意的三维光滑道路表面。另外，你可以在原有光滑路面上设置一些三维的障碍，如：台阶、凹坑、斜坡或拱形路面。你可以使用三维路面和任何 ADAMS/Tire 中的轮胎模型。使用光滑路面和任何操纵性轮胎模块或其它更高级的 FTire 或 SWIFT-Tyre 可以处理路面障碍以进行平顺性能、舒适性能和耐久性能的分析。

ADAMS/Tire 模块

■ 其它特殊的轮胎模块

- ◆ 除了 ADAMS/Tire 操纵性轮胎模块之外，ADAMS/Tire 还提供其它特殊的轮胎模块：
 - Pacejka Motorcycle Tire Model
A Pacejka tire model suitable for motorcycle handling analysis that can describe the tire-road interactions forces with tire-road inclination angles up to 60 degrees.
 - ADAMS/Tire FTire Module
FTire can describe the 3D tire dynamic response up to 120 Hz and beyond, due to its flexible ring approach for the tire belt. It can handle any road obstacle.
 - ADAMS SWIFT-Tyre Module
SWIFT-Tyre adds the tire dynamics of a rigid ring to a Pacejka tire model resulting in a tire model that can describe tire-road interaction during handling and ride and comfort applications up to 60 Hz (3D) accurately.
- ◆ 所有的轮胎模块都支持 ADAMS/Linear 的功能。

ADAMS/Tire 模块

■ ADAMS/Tire 模块的特色

- ◆ 下表列出了 ADAMS/Tire 中各轮胎模块的特色。

ADAMS/Tire Modules:	Features:	Requirements:
ADAMS/Tire Handling	Fiala Tire Model Pacejka '89* Tire Model Pacejka '94* Tire Model Pacejka 2002 Tire Model UA-Tire Tire Model 2D Road Models ----- 5.2.1 Tire and Road Models - Equation Method - Interpolation Method - Point Follower - Equivalent Plane	Full Simulation Package
ADAMS/Tire 3D Contact (was ADAMS/Tire Durability)**	Fiala Tire Model 3D Contact Road Model	Full Simulation Package
ADAMS/Tire 3D Road	Fiala Tire Model 3D Smooth Road and Road perturbations	Full Simulation Package
ADAMS/Tire FTire	2D FTire Model 3D FTire Model 2D Road Models	Full Simulation Package
ADAMS/Tire Swift	SWIFT-Tyre Model 2D Road Models	Full Simulation Package
ADAMS/Tire Motorcycle Tire	Pacejka Motorcycle Tire Model 2D Road Models	Full Simulation Package

ADAMS/Tire 模块

■ 你应该使用什么类型的轮胎？

- ◆ 每个轮胎模块在一个特定的应用领域是正确的。超出这一领域可能会出现不真实的分析结果。下图所示为每种类型的轮胎的正确适用领域，而下页的表表明不同的轮胎模型的最佳应用情况。

		transient	gyroscopic effect	scaling factors	combined slip	camber effects	co-ordinate system	max valid frequency (Hz)	tyre enveloping effects
ADAMS/ Handling Tire	PAC2002	Y	Y	Y	Y	Y	ISO	8	N
	PAC89	N	N	limited	ellips	limited	SAE/ISO	0.5	N
	PAC94	N	N	limited	ellips	Y	SAE	0.5	N
	FIALA	N	N	N	N	N	SAE	0.5	N
	5.2.1.	N	N	N	ellips	Y	SAE	0.5	N
	UA Tire	Y	N	N	ellips	Y	SAE	8	N
Specific Models	PAC_MC	Y	Y	Y	Y	Y	ISO	8	N
	SWIFT-Tire	Y	Y	Y	Y	Y	ISO	60	Y
	Ftire	Y	Y	N	Y	Y	ISO	120	Y

ADAMS/Tire 模块

- ◆ 通常来讲，ADAMS/Tire 的操纵性轮胎模块只适用于相当光滑的路面：路面障碍的波长不应当小于轮胎的周长。如果障碍的波长小于轮胎的周长，你应该使用 SWIFT-Tyre 或 FTire 轮胎模型以考虑轮胎周边变形的非线性的影响。
- ◆ 操纵性轮胎模块适用于描述轮胎的一阶响应，但是不考虑轮胎自身的特征频率的影响。因此，操纵性轮胎模块适用于在 大约 8 Hz 频率以下的响应分析，超过这一频率，轮胎模型就应该考虑轮胎周边橡胶带的影响，SWIFT 和 FTire 轮胎模型就考虑了其影响。

	ADAMS/ Handling Tire						Specific Models		
	PAC2002	PAC89	PAC94	FIALA	5.2.1.	UA Tire	PAC_MC	SWIFT-Tire	FTire
standstill	0	-	-	-	-	-	0	0	0
steady-state cornering	+	0	+	0	0	0	+	+	0
cornering over bumpy road	0	-	-	-	-	0	0	+	0
lane change	+	0	0	0	0	0	+	+	0
ABS braking	0	-	-	-	-	-	0	+	+
braking/power-off in a turn	+	0	0	0	0	0	+	+	0
same on bumpy road	0	-	-	-	-	-	0	+	+
shimmy	0	-	-	-	-	0	0	+	+
ride/comfort	-	-	-	-	-	-	-	+	+
chassis control systems > 8 Hz	-	-	-	-	-	-	-	+	+
chassis control with ride	-	-	-	-	-	-	-	+	+
durability	-	-	-	-	0	0	-	0	0
	-	not possible/not realistic							
	0	possible							
	+	best							

Tire 模型

■ 轮胎数据文件

- ◆ 在仿真过程要得到准确的轮胎力，轮胎数据是非常重要的。如果使用 Fiala 轮胎模型，你可以手工生成轮胎特性文件。如果你使用其它类型的轮胎，你必须使用适当的途径以得到轮胎特性文件所需要的各项系数。通常的方式是对实际的轮胎进行测试获得。除非你想使用的轮胎是已经做过试验，否则，一定要进行轮胎的试验，以得到轮胎特性文件必要的的数据。

路面类型

- 二维路面的接触采用 point-follower 的方法 (类似于一张平面的圆盘)。下面为使用这种方法所用的不同的路面类型 (主要是对应 FTIRE) :
 - ◆ DRUM - Tire test drum (requires a zero-speed-capable tire model).
 - ◆ FLAT - Flat road.
 - ◆ PLANK - Single plank perpendicular, or in oblique direction relative to x-axis, with or without bevel edges.
 - ◆ POLY_LINE - Piece-wise linear description of the road profile. The profiles for the left and right track are independent.
 - ◆ POT_HOLE - Single pothole of rectangular shape.
 - ◆ RAMP - Single ramp, either rising or falling.
 - ◆ ROOF - Single roof-shaped, triangular obstacle.

路面类型

- ◆ SINE - Sine waves with constant wave length.
 - ◆ SINE_SWEEP - Sine waves with decreasing wave lengths.
 - ◆ STOCHASTIC_UNEVEN - Synthetically generated irregular road profiles that match measured stochastic properties of typical roads. The profiles for left and right track are independent, or may have a certain correlation.
- ADAMS/Car中各种类型路面的样板文件在标准的ADAMS/Car 数据库下：
 - ◆ install_dir/acar/shared_car_database.cdb/roads.tbl/
 - ADAMS/Tire中各种类型路面的样板文件在：
 - ◆ install_dir /solver/atire/
 - ADAMS/Chassis中各种类型路面的样板文件在：
 - ◆ install_dir /achassis examples/rdf/

2D 路面的例子 (RAMP)

```
$-----MDI_HEADER
[MDI_HEADER]
FILE_TYPE      = 'rdf'
FILE_VERSION   = 5.0
FILE_FORMAT    = 'ASCII'
(COMMENTS)
{comment_string}
'ramp style road description'
$-----UNITS
[UNITS]
MASS           = 'kg'
LENGTH        = 'mm'
TIME          = 'sec'
ANGLE         = 'degree'
FORCE         = 'newton'
$-----MODEL
[MODEL]
METHOD        = '2D'
ROAD_TYPE     = 'ramp'
$-----PARAMETERS
[PARAMETERS]
OFFSET        = 0
ROTATION_ANGLE_XY_PLANE = 180
MU            = 1.0
$
HEIGHT       = 50
START        = 1000
SLOPE        = 1
```

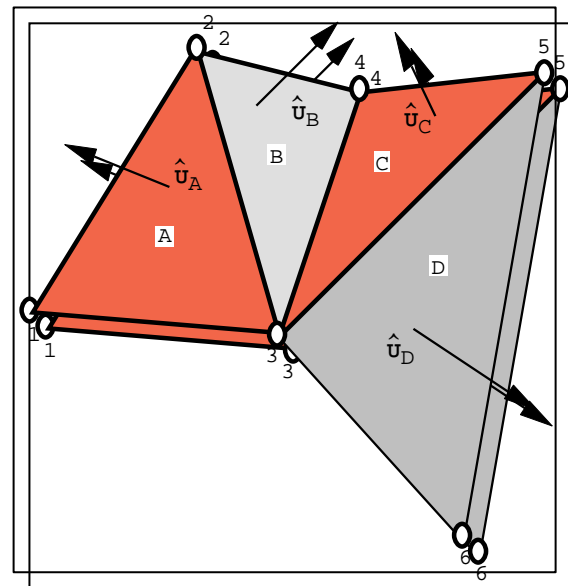
使用 3D 等效体积路面模型

- ◆ 3D 等效体积模型为三维的轮胎-路面接触模型，用来计算路面和轮胎之间交叉的体积。路面是用一系列离散的三角形片来表示，而轮胎则用一系列的圆柱表示。采用此路面模型，你可以模拟车辆在运动过程中碰到路边台阶、凹坑或在粗糙路面或不规则路面上运动的情形。

使用 3D 等效体积路面模型

■ 关于3D 等效体积路面模型

- ◆ 3D 等效体积路面模型为一般的三维表面，并用一系列的三角形片表示。右侧的图表示一个由编号为 1 到 6 的六个节点构成的路面表面。六个节点共构成四个三角形的面单元，分别表示为 A、B、C 和 D。每个三角形单元的向外的单位法向矢量如图所示。与有限元中网格的定义习惯非常相似，ADAMS/Tire 在定义路面时需要你首先指定每个节点在路面的参考坐标系下的坐标，然后再按照顺序指定由三个节点构成的三角形单元，对应每个单元，你可以指定不同的摩擦系数。



使用 3D 等效体积路面模型

■ 定义 3D 等效体积路面模型

- ◆ 你需要使用一个路面特性文件来定义三维路面。路面特性文件包含五个数据块：Header、Units、Model、Nodes和Elements。这些数据块可以按照任意的顺序出现在文件中，并且关键字可以放在其所属的数据块的任何顺序。

3D等效体积路面模型的例子

```
$-----MDI_HEADER
[MDI_HEADER]
FILE_TYPE      = 'rdf'
FILE_VERSION   = 5.0
FILE_FORMAT    = 'ASCII'
(COMMENTS)
{comment_string}
'flat 3d contact road for testing purposes'
$-----UNITS
[UNITS]
LENGTH        = 'mm'
FORCE          = 'newton'
ANGLE         = 'radians'
MASS           = 'kg'
TIME          = 'sec'
$-----DEFINITION
[MODEL]
METHOD         = '3D'
$-----OFFSET
[OFFSET]
X              = 0.0
Y              = 0.0
Z              = 0.0
$-----NODES
[NODES]
NUMBER_OF_NODES = 4
{node          x_value      y_value      z_value
1              1000000.0     50000.0     0.0
2              1000000.0     -50000.0    0.0
3              -1000000.0     -50000.0    0.0
4              -1000000.0     50000.0     0.0
$-----ELEMENTS
[ELEMENTS]
NUMBER_OF_ELEMENTS = 2
{node_1        node_2        node_3        mu
1              2              3              1.0
1              3              4              1.0
```

3D 光滑路面

- 3D 光滑路面允许你模拟很多类型的光滑路面，如停车场、比赛道路等等。一条光滑路面的涵义是指其曲率小于轮胎的曲率。
- ADAMS/Tire 中存储3D 光滑路面文件在标准的 TeimOrbit 格式路面数据文件 (.rdf).

路面数据文件的实例

```
$-----MDI_HEADER
[MDI_HEADER]
FILE_TYPE = 'rdf'
FILE_VERSION = 5.00
FILE_FORMAT = 'ASCII'
(COMMENTS)
{comment_string}
'Example of 3d spline road'
$-----UNITS
[UNITS]
LENGTH = 'meter'
FORCE = 'newton'
ANGLE = 'radians'
MASS = 'kg'
TIME = 'sec'
$-----DEFINITION
[MODEL]
METHOD = '3D_SPLINE'
$-----ROAD_PARAMETERS
[GLOBAL_PARAMETERS]
CLOSED_ROAD = 'NO'
SEARCH_ALGORITHM = 'FAST'
ROAD_VERTICAL = '0.0 0.0 1.0'
FORWARD_DIR = 'NORMAL'
MU_LEFT = 0.5
MU_RIGHT = 0.5
WIDTH = 5.000
BANK = 0.0
$-----DATA_POINTS
[DATA_POINTS]
{ X          Y          Z          WIDTH BANK MU_LEFT MU_RIGHT }
12.50000E+00  4.60432E-15  0.00000E-00  7.000  0.000  0.900  0.900
10.50000E+00  4.60432E-15  0.00000E-00  7.000  0.000  0.900  0.900
5.50000E+00   4.60432E-15  0.00000E-00  7.000  0.000  0.900  0.900
0.50000E+00   4.60432E-15  0.00000E-00  7.000  0.000  0.900  0.900
1.53081E-18   1.42109E-17  0.00000E-00  7.000  0.000  0.900  0.900
-2.50000E+00  4.68958E-16  0.00000E-00  7.000  0.000  0.900  0.900
-5.00000E+00  9.37916E-16  0.00000E-00  7.000  0.000  0.900  0.900
-7.50000E+00  1.39266E-15  0.00000E-00  7.000  0.000  0.900  0.900
-1.00000E+01  1.84741E-15  0.00000E-00  7.000  0.000  0.900  0.900
-1.25000E+01  2.30216E-15  0.00000E-00  7.000  0.000  0.900  0.900
-1.50000E+01  2.77112E-15  0.00000E-00  7.000  0.000  0.900  0.900
-1.75000E+01  3.22586E-15  0.00000E-00  7.000  0.000  0.900  0.900
-2.00000E+01  3.69482E-15  0.00000E-00  7.000  0.000  0.900  0.900
$-----END_DATA_POINTS
```

使用 3D 光滑路面

- ◆ 你可以象使用数据库下其它的 .rdf 一样参考 3D 光滑路面。另外，ADAMS/Car 中包含 3D 路面分析模式，称作 3D Smooth Road。从动画显示上考虑，路面的几何外形自动的生成。
- ◆ 使用DriverLite 的驾驶控制文件(.dcf文件)的例子如下所示：

```
[3D_SMOOTH_ROAD]
(STEERING)
ACTUATOR_TYPE = 'ROTATION'
METHOD = 'MACHINE'
(THROTTLE)
METHOD = 'MACHINE'
(BRAKING)
METHOD = 'MACHINE'
(GEAR)
METHOD = 'MACHINE'
(CLUTCH)
METHOD = 'MACHINE'
(MACHINE_CONTROL)
STEERING_CONTROL = 'FILE'
DCD_FILE_STEERING = 'mdids://shared/roads.tbl/mdi_3d_smooth_road.rdf'
SPEED_CONTROL = 'lon_accel'
LON_ACCEL_TARGET = 0.0
START_TIME = 1.0
MIN_ENGINE_SPEED = 1000.0
MAX_ENGINE_SPEED = 7000.0
```

如何使用 ADAMS/Tire

- 你可以生成你自己的轮胎模型，当然也可以使用 ADAMS/Tire 里所带的轮胎模型。下面介绍如何使用 ADAMS/Tire。



更多关于如何生成你自己的轮胎模型，参见 ADAMS/Tire 的在线帮助。

- 要使用 ADAMS/Tire：
 1. 定义轮胎。你如何定义轮胎，与你使用 ADAMS 的什么产品有关 (ADAMS/Chassis、ADAMS/Car 或 ADAMS/Solver)。
 - ◆ 不考虑你所使用的产品，你可以直接考虑使用 ADAMS/Solver，因为 ADAMS 的前处理模块都需要生成 MSC.ADAMS 的模型描述文件 (.adm 文件)，其中，与车辆的其它卡片描述一样，包含表示你模型中轮胎的必要的卡片描述。诸如：轮胎、悬架和起落架的支架等。每个轮胎最主要的描述包括一个作用在你悬架上轮胎的 GFORCE。

如何使用 ADAMS/Tire

2. 参考已有的轮胎特性文件：

- ADAMS/Tire (ADAMS/Car 共享数据库)
- 轮胎制造商或轮胎试验公司。
- 你自己生成的文件。例如：简单的轮胎，如 Fiala 轮胎模型，你可以手工生成。

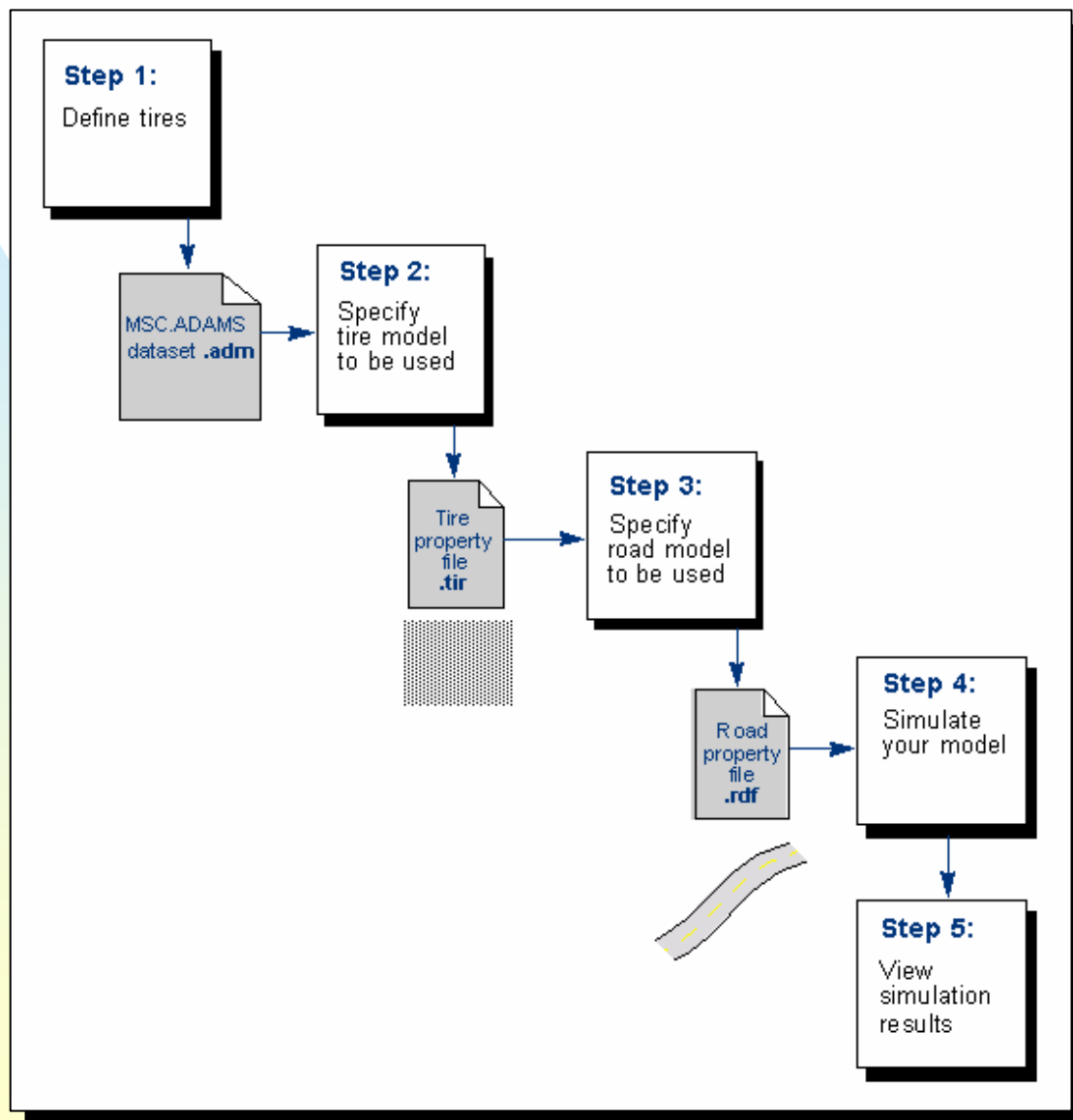
轮胎的特性文件指定 ADAMS/Tire 中所使用的轮胎的类型以及计算轮胎力和力矩特性的数据。数据量和数据类型依据所使用的轮胎类型而相应变化。在 MSC.ADAMS 的模型描述文件 .adm 中，一个 STRING 卡片描述中指定轮胎的特性文件。

3. 参考已有的路面特性文件。

路面特性文件中包含定义道路表面的数据和摩擦系数。路面可以使碰到，又可以是由三角形所表示的三维路面。在 MSC.ADAMS 的模型描述文件 .adm 中，一个 STRING 卡片描述中指定路面的特性文件。

4. 运行仿真你的模型。

如何使用 ADAMS/Tire



练习 12 建立一个轮胎的模板

- 在本练习中，你将生成一个轮胎的模板，其中包括如何设置你的轮胎模型。



本练习大约需要一个小时完成。

练习 12 建立一个轮胎的模板

■ 生成一个模板

要生成一个模板：

1. 从 **File** 菜单下选择 **New**。
2. 给模板起名 **wheel**。
3. 设置 **Major Role** 为 **wheel**。
4. 选择 **OK**。

练习 12 建立一个轮胎的模板

■ 生成信息交换器

要生成信息交换器：

1. 生成两个左/右对称的输入信息交换器作为前束角和外倾角（toe 和 camber）。它们应该是 **parameter real** 类型，并且其名字为 **toe_angle** 和 **camber_angle**。
2. 生成左/右对称的输入信息交换器，命名为 **wheel_center**，该输入信息交换器接受轮胎中心的位置。此生成信息交换器的位置为悬架中轮胎的定位点。对应的输出信息交换器为 **wheel_center**，在你前面练习 9 — 模板建模所生成的 MacPherson 悬架中。

*提示：*输入轮胎中心的初始值，否则的话，按照缺省的值，位置将为 0,0,0，这样的话，轮胎的中心将重叠在一起，下一步你将看到这一结果。记住左侧的 Y 方向的值为负。

练习 12 建立一个轮胎的模板

■ 生成构造坐标系和装配部件

要生成构造坐标系和装配部件：

1. 生成左/右对称的构造坐标系，命名为 **wheel_center**，用来作为转轴。其位置使用 **wheel_center** 输入信息交换器，与悬架的输出信息交换器相匹配。方向使用两个信息交换器前束角和外倾角来定义。(参见不同的方位设置选项)。检查一下轮轴的方向定义的是否正确，其 Z 轴应该指向车辆的外侧。(软件自动的生成)。
2. 对车轮，生成装配部件以连接到悬架上，命名其为 **wheel_to_susp**。参考坐标系应该为轮胎中心的构造坐标系。

练习 12 建立一个轮胎的模板

■ 生成轮胎部件：

要生成轮胎部件：

1. 从菜单 **Build** 下选择 **Wheel** 再选择 **New**。

2. 按照下面的参数生成轮胎部件：

- **Name:** wheel
- **Mass:** 20.0 (kg)
- **Ixx Iyy:** 5E4 (kg-mm**2)
- **Izz:** 1E4 (kg-mm**2)
- **Property File:** mdi_tire01.tir
- **Coordinate Reference:** cfl_wheel_center
- **Location:** 0, 0, 0 mm
- **Construction Frame:** cfl_wheel_center
- **Orientation:** 0, 0, 0 (deg)

3. 选择 **OK** 以生成轮胎部件。

注意：ADAMS/Car 自动地生成一对轮胎，并且按照轮胎特性文件生成轮胎的几何外形。

练习 12 建立一个轮胎的模板

■ 观察轮胎的几何外形：

要观察轮胎的几何外形：

1. 要观察轮胎的特性文件，打开修改轮胎的对话框，并选择**View File** 工具按钮。
2. 搜索名为 **DIMENSION** 的数据块。
在此数据块中的参数指定轮胎的几何外形的尺寸。

■ 连接装配部件和轮胎部件：

要连接装配部件和轮胎部件：

- ◆ 在 **cfl_wheel_center** 处装配部件和轮胎部件之间生成固定副。

练习 12 建立一个轮胎的模板

■ 修改信息交换器

- ◆ 修改 MacPherson 悬架中的装配部件输出信息交换器，以使两个模板之间能够正确的装配起来。

■ 要修改信息交换器：

1. 打开你在前面练习 9 中所生成的 MacPherson 悬架模板。确认使用你自己建的模板而不是标准数据库下的模板，因为二者之间有着不同的拓扑结构关系。
2. 从菜单 **Build** 下选择 **Communicator** 再选择 **Output**，然后选择 **Modify**。
3. 修改 **co[l,r]_suspension_mount** 以包含匹配的名字 **wheel_to_susp**。

此操作使 MacPherson 悬架模板可以传递给轮胎模板中轮胎应该与哪个部件相连的信息。

练习 12 建立一个轮胎的模板

■ 测试信息交换器

要测试信息交换器：

- ◆ 测试信息交换器并选择此轮胎模板和你要使用的悬架模板（确认两个目标均已打开）。

注意此轮胎模板是专门为你在练习 9 中所生成的 MacPherson 悬架模板而作的。如果你使用其它的悬架模板，你还需要保证其它的信息交换器也能够匹配。

练习 12 建立一个轮胎的模板

■ 测试装配

要测试装配

1. 存储模板以便进行一般的装配。
2. 切换到标准建模模式，并基于你的模板生成子系统。
3. 存储你上一步所生成的子系统。
4. 生成一般装配 (只用于测试目的，因为你未在装配中包含试验台)：
 - ◆ **Tools → Command Navigator → acar → files → assembly → new → generic**
 - ◆ **assembly class: generic**
 - ◆ **subsystems: wheel 和 macpherson**
5. 看看装配过程是否有问题。

练习 12 建立一个轮胎的模板

- ◆ **注意：**你刚生成的轮胎模板包含运行ADAMS/Car 和 ADAMS/Tire模块最基本的定义。你可以打开共享数据库下的模板 `_handling_tire.tpl` 可以看到更全面的轮胎的定义，只有这样，才可以使用 ADAMS/Car 中所有的分析功能 (如，quasi-static setup)，除非你需要一个特殊的模板。在模板 `_handling_tire.tpl` 其它的设置有：
 - 一对 DIFFs，对应于每个轮胎。静力学分析使用 DIFFs 确定轮胎的初始纵向滑移率，轮胎的 UDE 参数参考此 DIFF。
 - 一对 JPRIMs (`jo[lr]per_sse_jprim`)，在静力学分析过程约束轮胎。该约束类型为垂直原始约束，用来约束轮胎和路面。轮胎的 UDE 参数参考此 JPRIMs。
 - 输出信息交换器 `co[lr]_tire_force`，用来输出左/右轮胎力的 `adams_ids` 给 powertrain 模板，用来计算轮胎的转速。

第 15 章 模板的调研

- 在你建立或修改模板之前，一个非常重要的地方就是你要知道模板是如何建立和组织的。这不仅仅包含模板中的部件之间是如何连接的（约束、驱动和载荷），还包括模板之间需要交换哪些信息。本章介绍调研模板的一些方法，使你能够使用并生成新的模板，并保证其与其它模板相兼容。

模板的调研

- 本章内容：
 - ◆ 调研模板
 - ◆ 理解模板
 - ◆ 关于 Database Navigator

模板的调研

■ 调研模板主要包括：


- ◆ 得到模板中部件的信息。一旦你理解了模板是由哪些部件组成以及它们之间的连接关系，你就能够修改部件，建立你独特的模板。
- ◆ 得到模板中信息交换器的信息。光是机械的理解模板是如何工作的还不够的，你必须还理解其与其它模板和试验台之间如何交换信息。这些有助于你正确的连接你的模板。

理解模板

■ 帮助你理解模板是如何建立的一些准则：

1. 在你要获得信息的元件上点击鼠标右键，然后指向该元件的名称，再选择 **Modify** 或 **Info**。
2. 你能够使用模板建模下的 Build 菜单可以按照类别列出所有元件的信息，以及它们的特性，如质量、位置、方向等等。欲知所有元件的信息时，可以显示 Entity Information 的对话框。从此对话框中，你能够选择你要知道元件的类别。在标准界面下，选择 **Tools** → **Entity Info**。

理解模板

3. 当请求信息交换器的信息时，会显示 Communicators Info 的对话框。使用 Communicator Info 对话框列出不同的模板和试验台的信息交换器的信息。
4. 要显示部件之间的连接，可以选择 **Tools → Highlight Connectivity**。
5. 要以文本格式的文件显示所有的部件、约束的信息，可以在状态栏内右键点击  再选择  或 。
6. 使用 Database Navigator 探究你的模板。

关于 DATABASE NAVIGATOR

■ 显示 Database Navigator

- ◆ 你可以采用下面任意一种方法显示 Database Navigator :
 - 从菜单 Tools 下选择 Database Navigator。
 - 从菜单 Edit 下选择修改的命令，比如 Modify等，当没有任何对象被选时，显示 Database Navigator。
 - 从 Edit 的弹出式菜单，使用 Info 找到要查看对象的信息。
 - 使用 Browse 命令，在对话框输入一个浏览对象的名字。

关于 DATABASE NAVIGATOR

- ◆ Database Navigator 有几种显示对象信息的模式。你可以设置 Database Navigator 使其处于浏览对象的状态或处于观察对象信息的状态 (如观察对象之间的联系和依存关系)。Database Navigator 根据你的命令只显示某一类对象，例如，如果你要给模型改名，它就只显示你的数据库下的所有模型。另外一方面，你要搜索数据库中任何的模型对象，它就显示所有类型的模型对象。你也可以设置 Database Navigator 的对象类型的过滤等。
- ◆ Database Navigator 显示在它们数据库谱系下的对象。

关于 DATABASE NAVIGATOR

■ 观察模型的拓扑关系

- ◆ 你可以使用 Database Navigator 显示有关你模型中部件的信息。你可以：
 - 观察模型的拓扑关系 – 当你要求关于模型的拓扑关系的信息时，ADAMS/View 确定模型中包含哪些约束以及约束连接哪些部件。有两种不同的方式你可以观察部件的连接关系：基于部件或基于约束。



有关这两种方式的更多信息，参见 ADAMS/View 的在线帮助。

- 图形化观察拓扑关系 – 在图形化观察拓扑关系模式下，Database Navigator 显示某一个被选择的部件以及其与其它部件的连接关系。连接为约束（兰线）或力（红线）的关系。，部件以圆圈起来。每次你选择一个 Database Navigator 左侧目录树的一个部件，图形化观察拓扑关系模式就将其置于中心位置，并显示其与其它部件的连接关系。

练习 13 调研并完成模板

- 本练习测试某些有关模板建模方面的一些内容，包括定义信息交换器和约束，并使用 database navigator 调研你的模板。
- 我们推荐，不管何时，主要有可能，你都是修改已有的模板，而不是生成一个新的。为了能够修改已有的模板并客户化，使其为我所用，你必须首先能够很好的理解它们。



本练习大约需要一个半小时完成。

练习 13 调研并完成模板

■ 定义你的模板

要定义你的模板：

1. 复制 **_steer_training.tpl** (从你的老师那里获得) 到你自己的数据库下的 **template.tbl** 目录 (例如：acar_training.cdb/templates.tbl)。
2. 在模板建模状态下打开 **_steer_training.tpl** 。
一个齿轮齿条转向模板显示在窗口内。该模板尚未完成：你还需要定义约束，以及信息交换器和装配部件。使用 Database Navigator 调研该模板，看看部件及其图标，并试着确定你还需要在模板里定义哪些东西。
3. 进行一些必要的改变以正确的定义你的模板。如果你相要给自己一些挑战，可以从下页的一般步骤开始定义你的模板，如果你有困难或更愿意检查你的工作，可以从后面的详细步骤开始定义你的模板，以确定你应该作哪些修改。

练习 13 调研并完成模板

■ 定义你的模板的一般步骤

要定义你的模板：

1. 约束转向杆的运动。
2. 约束方向盘与转向杆之间的运动，这样方向盘的转向运动会带动转向杆的运动。(提示：使用齿轮副。有关齿轮副的帮助，选择对话框并按 **F1**键)。
3. 约束转向轴与转向拉杆架之间的运动。
4. 约束转向拉杆和转向架之间的运动。
5. 约束转向轴与转向拉杆之间的运动。(提示：使用齿轮副)。
6. 为拉杆架生成装配部件并命名其为 **rackhousing_to_body**。
7. 确认拉杆能够与 与你前面所生成的 MacPherson 悬架模板相连接。必要的话，生成任何的装配部件或信息交换器。

练习 13 调研并完成模板

8. 检查一下转向杆架能够正确装配，必要的话，生成任何的装配部件或信息交换器。
9. 生成名为 **steering_rack_joint** 的输出信息交换器，该信息交换器提供齿条约束的名称，这样的话，ADAMS/Car 内在的试验台能够正确地施加驱动。
10. 如果时间允许的话，在模板中增加位置信息交换器以定位齿条的端点该转向杆的端点位置。

练习 13 调研并完成模板

■ 定义你的模板的详细步骤

要定义你的模板：

1. 生成两个虎克绞 (万向轴) 约束：一个是在转向杆(steering column) 和中间轴(intermediate shaft) 之间，另一个是中间轴(intermediate shaft) 和转向轴(steering shaft)之间。
2. 生成减速齿轮副，以约束转向轴的回转副和转向杆的圆柱副，此齿轮副应该始终是激活的，传动比为 1，意味着方向盘的转动等速的传递到转向杆的转动。 __MDI_SDI_TESTRIG 需要一个回转副，这也是此模型是用在转向模板中的原因。
3. 在转向轴和转向拉杆架之间生成一个回转副。
4. 在转向拉杆和转向拉杆架之间生成一个移动副。

练习 13 调研并完成模板

5. 生成一个减速齿轮副，用来约束转向轴的回转运动和转向拉杆的移动运动。(传动比为 0.05)。
6. 生成一个装配部件，命名为 **rackhousing_to_body**，该装配部件将连接到车身上。将其置于 **cfs_rack_mount** (为了生成一个单个的装配部件)，并将其与转向拉杆架使用固定副固结在一起。这样的话，转向拉杆将固定到车身上。
7. 打开前面生成的 MacPherson 悬架模板，并检查装配部件和在斜拉杆的装配信息交换器。在 MacPherson 悬架模板中的装配输入信息交换器 **tierod_to_steering** 需要模板 **steer_training** 中定义的输出装配信息交换器 **tierod_to_steering** 进行装配。该输出装配信息交换器输出部件转向拉杆 **rack**。
8. 生成一个左/右 对称的输出信息交换器以输出单个的部件 **ges_rack**。注意：转向杆是固接到装配部件 **steering_column_to_body** 上的。

练习 13 调研并完成模板

9. 编辑车身模板，以保证信息交换器能够正确的匹配。
10. 生成一个单个的 entity 类型的输出信息交换器 **joint_for_motion**，并命名其为 **steering_rack_joint**，该输出信息交换器输出前面第 4 步定义的转向拉杆的约束。该信息交换器将提供约束的名字，以便 ADAMS/Car 内置的试验台能够在该约束上施加驱动。



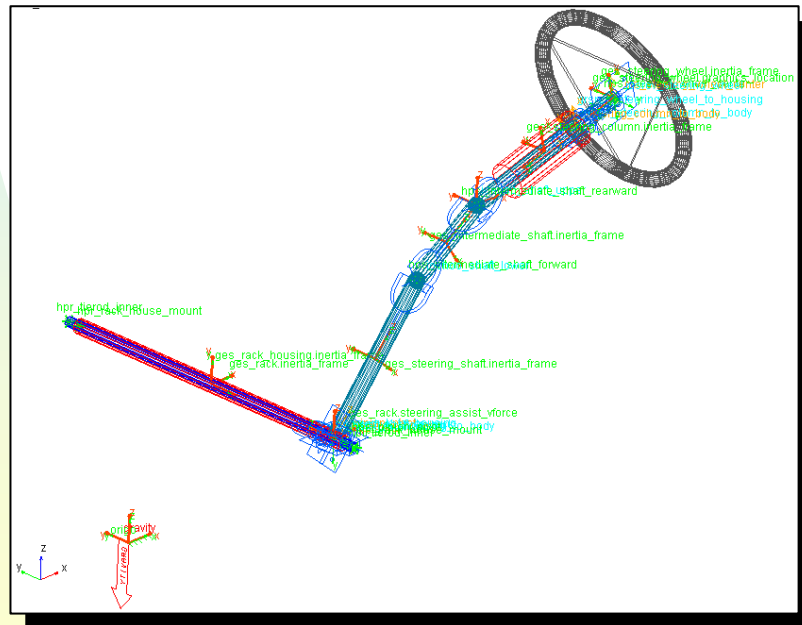
有关试验台中所用的信息交换器的详细信息，参见 ADAMS/Car 在线帮助手册 Templates tab 部分或直接调研试验台模板本身。

你的模板看起来类似于 `steer_final.tpl`，并用于最终整车的分析中。

练习 13 调研并完成模板

11. 如果时间允许的话，在模板中增加位置输出信息交换器输出转向拉杆的位置以定位斜拉杆的端点。参见 KBA 9184：

[http://support.adams.com/kb/faq.asp?ID=kb9184.dasp.](http://support.adams.com/kb/faq.asp?ID=kb9184.dasp)



练习 13 调研并完成模板

- 下面列出在本练习前后所有的信息交换器。(欲取得此信息，选择 **Build** → **Communicator** → **Info**)。
- 修改前：

Listing of input communicators in '_steer_training'

Communicator Name: Class: From Minor Role: Matching Name:

cis_steering_column_to_body mount inherit steering_column_to_body

1 input communicator was found in '_steer_training'

Listing of output communicators in '_steer_training'

Communicator Name: Class: To Minor Role: Matching Name:

cos_max_rack_displacement parameter_real inherit max_rack_displacement

cos_max_rack_force parameter_real inherit max_rack_force

cos_max_steering_angle parameter_real inherit max_steering_angle

cos_max_steering_torque parameter_real inherit max_steering_torque

cos_steering_wheel_joint joint_for_motion inherit steering_wheel_joint

5 output communicators were found in '_steer_training'

练习 13 调研并完成模板

■ 修改后：

Listing of input communicators in '_steer_final'

Communicator Name: Class: From Minor Role: Matching Name:

cis_rackhousing_to_body mount inherit rackhousing_to_body

cis_steering_column_to_body mount inherit steering_column_to_body

2 input communicators were found in '_steer_final'

Listing of output communicators in '_steer_final'

Communicator Name: Class: To Minor Role Matching Name:

co[lr]_tierod_to_steering mount inherit tierod_to_steering

cos_max_rack_displacement parameter_real inherit max_rack_displacement

cos_max_rack_force parameter_real inherit max_rack_force

cos_max_steering_angle parameter_real inherit max_steering_angle

cos_max_steering_torque parameter_real inherit max_steering_torque

cos_steering_rack_joint joint_for_motion inherit steering_rack_joint

cos_steering_wheel_joint joint_for_motion inherit steering_wheel_joint

8 output communicators were found in '_steer_final'

练习 13 调研并完成模板

■ 在 ADAMS/View 中测试你的转向模板

注意：一般来说，在 ADAMS/View 进行仿真可能会不能正常工作，而且 ADAMS/View 的模型也不能无缝地与 ADAMS/Car 模型转换。因此，一个单个的模板能够在 ADAMS/View 中进行测试，但是，在 ADAMS/View 中的其它操作通常情况下都不能正常工作。因此在 ADAMS/View 下对 ADAMS/Car 的模型进行操作一定要小心。

■ 要测试该模板：

1. 从菜单 **Tools** 下选择 **Command Navigator**。
2. 选择 **acar interface switch aview**，此命令将切换到 ADAMS/View 的界面下。注意：你也可以通过修改你自己的 .acar.cfg 文件增加环境变量的方法直接在 Tools 菜单下增加一个切换到 ADAMS/View 的选项。



详细信息，参见 ADAMS/Car 在线帮助手册中 Configure 部分。

练习 13 调研并完成模板

3. 在方向盘的回转副上增加驱动 (如果你不是很熟悉 ADAMS/View , 请与指导老师联系)。
4. 从菜单 **Settings** 下指向 **Solver** , 再选择 **Solver Executable**。
5. 设置 **Executable** 为 **Internal [default]**。
6. 设置 **Category** 为 **Display**。
7. 设置 **Update Graphics** 为 **At Output Step**。
8. 选择 **Close**。
9. 运行一次 **5-second**、**200-step** 的仿真。

你应该看到方向盘的运动传递到转向拉杆上的移动运动。如果没有的话, 或出现了错误的话, 切换回 ADAMS/Car 模板建模状态 (**Tools** → **Select Mode** → **ADAMS/Car Template Builder**)。

练习 13 调研并完成模板

10. 删除你在第 3 步中所加的驱动，因为该驱动是不需要的，它与 ADAMS/Car 的分析相矛盾。
11. 从菜单 **Settings** 下选择 **Solver** 再选择 **Solver Executable**。
12. 设置 **Executable** 为 **External**。
13. 选择 **Close**。
14. 从菜单 **Tools** 下选择 **Select Mode** 再选择 **Switch to A/Car Template Builder**。

练习 13 调研并完成模板

- 调研模板的一些提示：

*注意：*因为本节中要在 Database Navigator 下进行很多步操作，因此首先确认你已经先打开了 Database Navigator (**Tools** → **Database Navigator**)。

- 调研模型的拓扑结构

- 要列出部件和连接的关系：

- ◆ 要查看部件和连接，设置在 Database Navigator 下部的选项按钮为 **Bodies**、**Constraints** 或 **Forces**。
- ◆ 设置上部的选项为 information。

练习 13 调研并完成模板

■ 要查看部件的拓扑关系：

1. 从对话框的顶部可选菜单选择 **Topology by Parts** 或 **Topology by Connections**。
2. 从查看窗口的目录树选择一个对象。
3. 该对象的拓扑关系以文本形式显示在右侧的窗口内。

■ 要观察部件的图形化拓扑关系：

1. 从对话框的顶部可选菜单选择 **Graphical Topology**。
2. 从查看窗口的目录树选择一个对象。
3. 该对象的拓扑关系以图形的形式显示在右侧的窗口内。

练习 13 调研并完成模板

■ 观察对象的相互关系：

- ◆ 你可以使用 Database Navigator 显示一个被选对象所相关的对象。
例如：你可以在列表树中选择一个约束以显示其所使用的 I 和 J 标记点。你又可以选择观察使用被选对象的对象。

■ 要观察对象的相互关系：

1. 从对话框的顶部可选菜单选择 **Associativity**。
2. 设置 associativity：
 - ◆ 要显示某个被选对象所使用的对象，选择 **Uses**。
 - ◆ 要显示使用被选择对象的对象，选择 **Is Used By**。
3. 从查看窗口的目录树选择一个对象。
4. 与该对象相关的对象以文本形式显示在右侧窗口内。

练习 13 调研并完成模板

- 要设置对象的自动导向：
 - ◆ 要观察某个对象被哪些对象所使用，选择 **Auto Navigate**。
- 要存储当前的相关性信息到一个文件中：
 - ◆ 选择 **Save to File**。
- 通过 Database Navigator 观察对象的信息
 - ◆ 你能够使用 Database Navigator 就像使用信息窗口一样显示有关被选对象的信息。
- 要显示对象信息：
 1. 从对话框的顶部可选菜单选择 **Information**。
 2. 从查看窗口的目录树选择一个对象。
 3. 有关该对象相关的信息以文本形式显示在右侧窗口内。

练习 13 调研并完成模板

- 要存储信息到一个文件中：
 - ◆ 选择 **Save to File**。
- 要列出信息交换器的类型：
 1. 从菜单 **Build** 下选择 **Communicator**，然后再选择 **Info**。
 2. 设置 **Type** 为 **Input**。
 3. 设置 **Entity** 为 **All**。
 4. 选择 **OK**。
- ◆ 要观察哪些对象依存一个信息交换器：
 1. 从菜单 **Tools** 下选择 **Database Navigator**。
 2. 在 **Filter** 文本栏内输入 **ci***。
 3. 从对话框的顶部可选菜单选择 **Associativity**。
 4. 要得到相关信息，选择 **Uses** 或 **Is Used By**。

练习 13 调研并完成模板

■ 位置和方位的参数化

- ◆ 要找出有关参数化的依存关系是如何设置的，你可以调研两类对象作为着手点：hardpoints 和 input communicators。
- ◆ Hardpoints 没有依存关系，因此，你可以将其作为参数化的着手点。你可以以 hardpoints 参数化构造坐标系和其它元件。
- ◆ 你可以使用参数变量参数化构造坐标系的位置和方位。



更多信息，参见 ADAMS/View 的在线帮助。

练习 13 调研并完成模板

■ 要查看哪些对象依存 hardpoint :

1. 在 **Filter** 文本栏内输入 **hp***。
2. 选择一个 hardpoint。
3. 选择 **Apply**。
出现信息窗口，显示 hardpoint 的位置坐标值。
4. 在 Database Navigator内，在对话框的顶部可选菜单选择 **Associativity** 以观察依存关系列表，同样，你可以选择 **Highlight** 以直接观察 hardpoint 的位置。
5. 你可以在某个构造坐标系下点击鼠标右键并选择 **Info** 或 **Modify**。
显示的信息包含模板中位置的参数化信息，对于构造坐标系，你可以查看其位置和方位的参数化表达式，以及哪些对象依存于这些构造坐标系。

练习 13 调研并完成模板

- 要查看哪些对象依存于构造坐标系：
 1. 在 **Filter** 文本栏内输入 **cf***。
 2. 设置对话框的顶部可选菜单选择 **Associativity** 。
- 要列出所有参数变量的依存关系：
 1. 在 **Filter** 文本栏内输入 **pv***。
 2. 要查看参数化变量，设置对话框的顶部可选菜单选择 **Information**。
 3. 要查看其依存关系，设置对话框的顶部可选菜单选择 **Associativity** 。

第 16 章 其它应用

- MSC.Software 提供一些与 ADAMS/Car 兼容的其它应用。下面介绍如何使用这些应用进行车辆的仿真。更多信息，参见相关模块的帮助文档。

其它应用

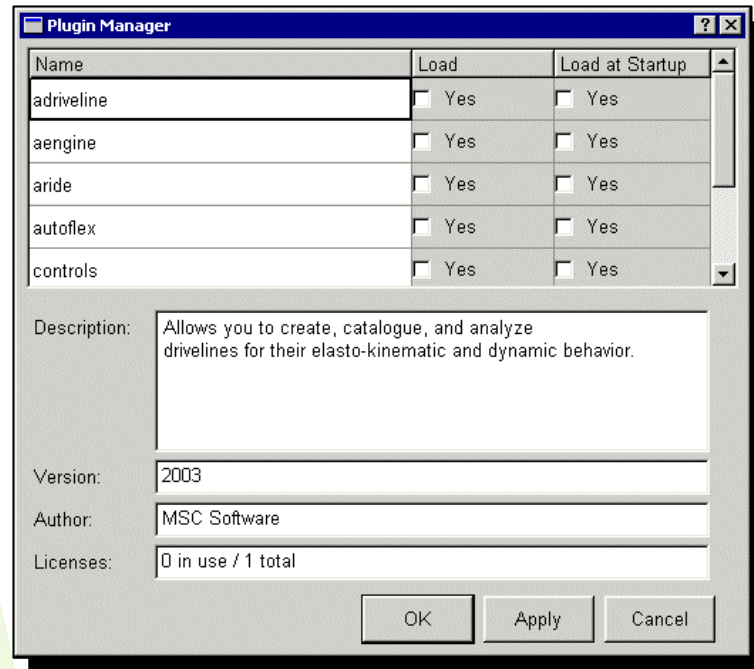
■ 本章内容：

- ◆ 即插即用功能
- ◆ ADAMS/Car Ride——平顺性分析模块
- ◆ ADAMS/Vibration——振动分析模块
- ◆ ADAMS/Durability——耐久性分析模块
- ◆ ADAMS/Driveline——传动系模块
- ◆ ADAMS/Linear——线性化模块
- ◆ ADAMS/Controls——控制模块
- ◆ ADAMS/Insight ——试验分析模块

即插即用功能

- 在 ADAMS 2003 发布后，增加了一个插件管理器。该插件管理器是一个有用的额外工具，使用插件管理器，你可以一次性地选择加载或卸载所有的插件。你可以在启动后选择加载插件，或每次启动都自动的加载插件。
- 要打开插件管理器：
 1. 启动 ADAMS/Car 标准界面。
 2. 从菜单 **Tools** 下选择 **Plugin Manager**。ADAMS/Car 显示 Plugin Manager，如下页所示。

即插即用功能



3. 选择下面选项已加载你想要的插件：
- ◆ **Load** – 在当前进程中加载插件。
 - ◆ **Load at Startup** – 每次启动过程都自动地加载某插件。
- 注意：**你需要相应插件的 license 许可。

即插即用功能

4. 选择 **OK**。

被选的插件开始加载。如果你退出或重启 ADAMS/Car，这些插件并不加载，除非你选择在每次启动时自动加载。

ADAMS/Car Ride——平顺性分析模块

- ADAMS/Car Ride 为 ADAMS/Car 的一个插件，可以用来评估功能化数字车辆的性能，从操纵性能到平顺性能和舒适性能，包括一些必须的元件、模型和事件定义，以便在平顺性频域进行建模、测试和后处理。



ADAMS/Car Ride——平顺性分析模块

- ADAMS/Car Ride 的使用方式是基于车辆虚拟样机的扩展，用于操纵性能分析同样的数据库，用于平顺性和舒适性方面的分析。ADAMS/Car Ride 使你能够交换元件的数学描述(阻尼器、衬套、液压悬挂及轮胎等)，这样可以适合你要研究的现象。详细的元件库包括：

ADAMS/Car Ride——平顺性分析模块

- ◆ 一个通用的基于 GSE的阻尼器用户定义元件 (UDE) ，适合于包含一个来自于使用 Mathworks RealTime Workshop 功能详细的阻尼器描述，举例来说，使用 ADAMS/Car Ride 推出的 simulink 模型能够包含此阻尼器的 UDE 模型。
- ◆ 详细的频率和幅值相关的液压悬挂模型，其数学模型描述是由德国的 Audi AG 公司提供的。该元件模型为单独统一的工具，自动地确定液压悬挂数学模型的参数，使其响应与用户指定的动态刚度和损失角特性相匹配。
- ◆ 详细的与频率相关的衬套模型。
- ◆ 一个更新的新版的 ADAMS/Tire SWIFT 轮胎模型，该轮胎模型允许“0”速度（不动的轮胎）和运动的路面（在试验台台架上）。此种轮胎的功能需要 ADAMS/Tire SWIFT 轮胎的 license。

ADAMS/Car Ride——平顺性分析模块

- ◆ 一个一般元件试验台，允许你对元件的性能进行单独的测试。该元件试验台可以进行从一个到六个通道的激励（三个移动的和三个转动的），使用或者预定义的驱动或者预定义的载荷。
- 一旦你详细的指定元件的特性，你能够对你的整车模型按照预定义的分析过程进行一系列的平顺性能和舒适性能的分析。使你能够评估典型的系统平顺性能矩阵。此分析是基于一个外加的四轮试验台，既支持分析类的激励信号（例如，正弦扫频信号），又支持测试类的激励信号（例如，RPC 格式的试验信号）输入。同时还提供路面的粗糙度模型，以模拟用户指定的功率谱密度 (PSD) 数据生成路面外形。

ADAMS/Car Ride——平顺性分析模块

- 对所有的仿真事件，ADAMS/Car Ride 支持时域和频域的逼近。时域的逼近，是基于 ADAMS/Solver，提供包括模型中非线性影响的精确分析结果，而频域逼近，是基于 ADAMS/Vibration，提供模态空间概念分析有效的工具。

ADAMS/Vibration——振动分析模块

- ADAMS/Vibration 为 MSC.ADAMS 的即插即用的工具，用于进行频域的分析。使用 ADAMS/Vibration，你可以研究你的 MSC.ADAMS 的强迫振动响应。例如，你能够仿真在颠簸路面上驱动一台车辆并测试其频率响应，输入和输出都是在频域描述的。
- 使用 ADAMS/Vibration，你可以：
 - ◆ 分析一个模型在不同的操作点处的频域内的强迫振动响应。

ADAMS/Vibration——振动分析模块

- ◆ 在频域分析包含液压系统、控制系统和用户的子系统。
- ◆ 从你的 MSC.ADAMS 产品的时域分析到 ADAMS/Vibration的频域分析的线性化转换完整而且快捷。
- ◆ 可以定制振动分析的输入和输出信号。
- ◆ 指定频域输入函数，如正弦扫频/扫幅、PSD或不平衡转子
- ◆ 生成用户定义的基于频率的载荷。
- ◆ 求解感兴趣频段内的系统模态。
- ◆ 评估频响函数，包括幅频特性和相频信号。
- ◆ 动画显示强迫振动和每一阶模态响应。
- ◆ 表格显示强迫振动每阶模态的贡献。
- ◆ 表格显示模型元件对系统模态的动态的、稳态的和耗散能量的贡献
- ◆ 指定运动输入。

ADAMS/Durability——耐久性分析模块

- ADAMS/Durability 使你能够更快捷轻松的工作，很容易地使用 PRC III格式文件的方式与疲劳试验设备进行交换数据，以及使用 DAC 文件格式借助疲劳寿命分析软件进行疲劳方面的分析，或者结合有限元软件进行应力恢复。
- ADAMS/Durability 提供下列好处：
 - ◆ 减少硬盘空间的需求，提供提供直接的 RPC III和 DAC 文件输入输出的功能提高仿真的效率。例如：你使用 MSC.ADAMS 进行仿真 25s，300 个数据通道，采样频率为 每秒 409.6 个数据点，以捕捉疲劳相关数据，MSC.ADAMS 需要大约 48 MB的数据量，而使用 RPC III 文件只要 6 MB。

ADAMS/Durability——耐久性分析模块

- ◆ 提供访问 ADAMS/View 或专业模块 ADAMS/Aircraft、ADAMS/Car、ADAMS/Engine 和 ADAMS/Rail 等系统级仿真产品。
- ◆ 提供使用 MSC.Nastran 或 ANSYS 进行动应力恢复的功能。
- ◆ 进行弹性体的模态应力恢复。
- ◆ 进行刚性体准静态应力恢复的功能。
- ◆ 提供使用 MSC.Fatigue 或 FE-Fatigue 进行部件寿命的预测的功能。

ADAMS/Driveline——传动系模块

■ ADAMS/Driveline

- ◆ ADAMS/Driveline 允许你建立并测试完整的传动系及其部件的功能化数字样机。ADAMS/Driveline 通过单独启用或在 ADAMS/Car 中作为插件的方式进行使用，单独启用 ADAMS/Driveline，类似于 ADAMS/Car 并可以进行同样的分析。
- ◆ 你能够使用ADAMS/Driveline生成传动系的虚拟样机，进行性能分析部件疲劳寿命评估或 NVH 的特性分析。你还可以将传动系导入到 ADAMS/Car 进行包括传动系在内的整车动力学分析。
- ◆ 类似于其它汽车模块，如 ADAMS/Car 和 ADAMS/Engine，ADAMS/Driveline 也有两种操作模式：
 - 标准界面(Standard Interface) – 允许设计人员和分析人员从可用的模板数据库中生成传动系装配，填入设计参数值再进行仿真。
 - 模板建模(Template Builder) – 允许专家用户修改或生成模板，以满足不同公司不同的需要。

ADAMS/Driveline——传动系模块

- ◆ ADAMS/Driveline 应用领域包括：
 - 传动系布置的概念性工具
 - 整车前驱、后驱或全驱配置情况下操纵性能分析。
 - 力矩传递和力矩分配研究。
 - 在系统级和部件级进行振动分析。
 - 部件疲劳寿命预测。
 - 控制系统设计和验证。
 - 传动系包络空间研究。
 - 在高操纵速度下次级运动的研究。
 - 部件弹性影响的研究。
- ◆ ADAMS/Driveline 模板库提供丰富的元件库，包括：离合器 (clutches)、弹性轴(flexible shafts)、synchronized gear pairs, bearings, viscous couplings, differentials, backlash, and gears.
- ◆ ADAMS/Driveline catalog of tests and analyses includes split mu, uphill and downhill driving, impulse, step, and ramp torques, tip-in tip-out, gear shift, rock cycle, and clutch misuse.

ADAMS/Linear——线性化模块

■ ADAMS/Linear

- ◆ ADAMS/Linear 的关键功能就是线性化非线性的 MSC.ADAMS 方程。得到的方程可以用来计算系统的自然频率（特征值）和模态振型（特征向量）。ADAMS/Linear 也同样有助于缩短控制系统的设计和 MSC.ADAMS 机械系统的设计之间的距离，因为你可以输出机械系统的空间状态矩阵（plant 模型），相对于测试系统的输出和控制信号的输入。

ADAMS/Controls——控制模块

■ ADAMS/Controls

- ◆ 控制算法现在可以应用到车辆模型。例如：悬架的自动控制车辆的操纵性设计，使其可以在湿滑的公路上避开障碍。控制系统设计师能够在计算机上将特定的车辆控制器与 MSC.ADAMS 模型提供的精确结果进行协调。ADAMS/Controls 允许设计师在计算机上看到每次改变的影响，甚至比较控制策略。而这一切在物理样机试验则是相当地昂贵费时。此接口允许你在 Matlab 或 EASY5 中模拟控制算法。

ADAMS/Insight —— 试验分析模块

■ ADAMS/Insight

- ◆ 在 ADAMS/Car 中使用 ADAMS/Insight，你可以策划并运行一系列的试验过程，以测试你悬架和车辆的性能。ADAMS/Insight 有助于你理解你的设计，区分出关键的或非关键的参数。另外，你将能够在用户要求的更广范围内进行设计决策，同时还可以通过考虑制造偏差进行鲁棒性设计。



更多关于在 ADAMS/Car 中使用 ADAMS/Insight，参见 ADAMS/Car 的在线帮助。

练习 14 在 CAR 中使用 LINEAR

- 在本练习中，你将学习如何在某个操作点上使用 ADAMS/Linear 分析整车的模态振型。
 - ◆ ADAMS/Linear 将在某个操作点上线性化你的车辆模型，并返回特征值和特征向量，这样有助于验证你的模型或调查你车辆中共振频率。例如，各阶模态的动画扫描有助于找出系统中的部件是否正确的连接。要得到特征值和特征向量，使用 ADAMS/Car 中所生成的 .acf 文件以提交 LINEAR 命令。在 ADAMS/View 中，能够直接在工具栏内进行此项分析，但在 ADAMS/Car 没有此界面，最直接的办法就是使用 .acf 文件和外部的 ADAMS/Solver。本练习的第一步就是生成 .acf 文件。



本练习大约需要半个小时完成。

练习 14 在 CAR 中使用 LINEAR

■ 生成 .acf 文件

要生成 .acf 文件：

1. 在 ADAMS/Car 中打开共享数据库下的 **MDI_Demo_Vehicle.asy**。
2. 从菜单 **Simulate** 选择 **Full-Vehicle Analysis** 再选择 **Straight-Line Behavior** 最后再选择 **Acceleration**。
3. 在加速仿真对话框中按照下列参数填入参数：
 - ◆ **Output Prefix:** linear_example
 - ◆ **End Time:** 5
 - ◆ **Number Of Steps:** 100
 - ◆ **Mode of Simulation:** files_only

练习 14 在 CAR 中使用 LINEAR

指定 files_only，因为你并不想运行仿真，这一步只是用来设置车辆模型并生成 .adm、.acf 和 .dcf 文件。在此，ADAMS/Car 将读入特性文件并存储相关数据在 .adm 文件中，因为当你通过 ADAMS/Car 的界面提交分析时，软件会自动的读入特性文件。如果你想手动加载特性文件，你可以使用 ADAMS/View 的命令 (acar analysis read property_files)。

- **Road Data File:** <shared>\roads.tbl\mdi_2d_flat.rdf
- **Initial Velocity:** 10 km/hr
- **Start Time:** 1
- **Open-Loop** Throttle
- **Final Throttle:** 50
- **Duration of Step:** 5
- **Gear Position:** 2
- **Steering Input:** locked

练习 14 在 CAR 中使用 LINEAR

4. 选择 OK。

ADAMS/Car 显示下面的信息：

```
Reading in property files...
Reading of property files completed.
Setting up the vehicle assembly for Driving Machine maneuver...
Setup of vehicle assembly completed.
Writing assembly information to ADAMS/Solver dataset...
ADAMS/Solver files written successfully.
ADAMS/Car writes the files to your current working directory. To check where
ADAMS/Car saved the files, go to File -> Select Directory).
To set up your .acf file, you simply insert the LINEAR solver command after your analysis,
specifying whatever options you want (see the LINEAR command in the ADAMS/Solver
online help). MSC.ADAMS will linearize about the operating point (the states) that it sees
at the end of the analysis. For example, here is the .acf file you just created:
file/model=linear_example_accel
preferences/solver = none
output/nosep
control/function=user(906,2,29,33,22,3,23,1)
!
stop
mine looks like this:
file/model=linear_example_accel
preferences/solver=F77
output/nosep
control/ routine=abgVDM:EventInit, function=user(3,1,8,0,2,5,7)
control/ routine=abgVDM:EventRunAll, function=user(0)
!
stop
```

练习 14 在 CAR 中使用 LINEAR

5. 编辑 **linear_example_accel.acf** 文件以包含特征值求解命令。

Your .acf file should look as follows:

```
file/model=linear_example_accel
preferences/solver = none
output/nosep
control/function=user(906,2,29,33,22,3,23,1)
linear/eigensol
!
stop
!
```

You are almost ready to run the linear analysis. First, you must make sure you save the proper output file. To save your mode shape results, you have to turn on the results file (.res), which is usually turned off for ADAMS/Car.

练习 14 在 CAR 中使用 LINEAR

6. 编辑 .adm 文件以包括 RESULTS/ 卡片在 .adm 文件的尾部。

```
!----- ANALYSIS SETTINGS -----  
!  
INTEGRATOR/  
, GSTIFF  
, ERROR = 0.01  
!  
OUTPUT/  
, REQSAVE  
, GRSAVE  
, NOPRINT  
!  
RESULTS/  
END
```

注意：要产生结果文件(.Res)，另一个办法就是显示 **Output Files** 的对话框 (**Settings** → **Solver** → **Output Files**)。在出现的对话框中，选择 **More**，并设置 **Output Category** 到 **Results (.res) Content**，然后选择 **Linear**。这些操作一定要在输出 .adm 文件之前进行。

练习 14 在 CAR 中使用 LINEAR

■ 使用外部的 solver 解算模型

要使用外部的 solver 解算模型：

1. 在命令输入窗口，输入启动 MSC.ADAMS 的命令：
 - ◆ Windows: **adams05r2**
 - ◆ UNIX: **adams05r2 -c**
2. 输入 ADAMS/Car 的选项 **acar**。
3. 要运行 ADAMS/Car solver，输入 **ru-solver**。

练习 14 在 CAR 中使用 LINEAR

4. 要输入你的 solver 的命令文件 **linear_example_accel.acf**。

注意：确认此文件以及其它所有相关文件都在你当前运行 ADAMS/Car 的目录内。

仿真开始，并产生输出文件，包括结果文件 .Res 文件，文件中包含特征值求解的结果。你也可以直接输入下面的命令：

- ◆ Windows: **adams05r2 acar ru-solver linear_example_accel.acf exit**
- ◆ UNIX: **adams05r2 -c acar ru-solver i linear_example_accel.acf exit**

练习 14 在 CAR 中使用 LINEAR

- 在 ADAMS/PostProcessor 中观察模态振型：
要观察模态振型：
 1. 启动 ADAMS/Postprocessor。
 2. 从菜单 **File** 下选择 **Import**, 再选择 **Analysis Files**。
 3. 在 **File Name** 对话框内，输入下面三个文件
linear_example_accel.gra、**linear_example_accel.res** 或
linear_example_accel.req 中的任意一个，将基于文件的根名读入所有的三个文件。

练习 14 在 CAR 中使用 LINEAR

4. 在 **Model Name** 文本栏内，点击鼠标右键并选择 **Model** 再选择 **Guesses** 然后再选择 **MDI_Demo_Vehicle**。
5. 选择 **OK**。
你的车辆模型加载到 ADAMS/PostProcessor 中。
6. 在空白区域点击鼠标右键并选择 **Load mode shape animation**。
7. 当你得到一条信息表明 animation will be deleted, 选择 **OK**。
8. 通过在 **Mode Number** 文本栏内输入感兴趣的模态的阶数观察相应该阶的模态。要查看相应的模态的特征值，选择 **Table of Eigenvalues**。例如，看看第 170 阶模态，并使用 play 工具动画显示该阶模态。

练习 14 在 CAR 中使用 LINEAR

- 有关 ADAMS/Solver 的背景信息
 - ADAMS/Solver 中的文件
 - ◆ ADAMS/Solver dataset files (.adm) – 定义模型中的元件如：部件、约束、载荷等等。
 - ◆ ADAMS/Solver command files (.acf) – 定义仿真过程中需要执行的命令。
 - 在单独的 ADAMS/Solver 中仿真
 - ◆ Interactive
 - ◆ Not scripted – 一个一个的输入命令
 - ◆ Scripted – 使用 ADAMS/Solver 的命令文件 (.acf)
 - ◆ Batch – 在后台使用ADAMS/Solver 的命令文件 (.acf)运行多个任务。
- 注意：*ADAMS/Solver 的命令文件中必须以要分析模型的名字开头，以 STOP 命令结束。

你也可以在 ADAMS/View 内运行外部的ADAMS/Solver 。

练习 15 整车装配

- 在本练习中，你将学习定制多个基本操控的方法，建立这些基本操控的适当的应用领域，并为每步基本操控设置终止条件。
- 本练习包含下面几节：
 - ◆ 定制一个事件
 - ◆ 生成三个基本操控
 - ◆ 定义每个基本操控的参数
 - ◆ 使用该事件运行一次分析



本练习大约需要一个小时完成。

练习 15 整车装配

- 定制一个事件
 - ◆ 在你生成一个事件之前，你必须先生成一个新文件，如下所示。
- 要显示事件发生器(Event Builder)：
 - ◆ 在 ADAMS/Car 中：
 - 打开装配 **MDI_DEMO_VEH**。
 - 从菜单 **Simulate** 下选择 **Full-Vehicle Analysis**，再选择 **Event Builder**。
 - ◆ 软件显示事件发生器(Event Builder)，其为灰色的，因为你尚未定制一个 .xml 文件。

练习 15 整车装配

■ 要生成一个新的事件文件：

1. 从事件发生器(Event Builder) 的菜单 **File** 下选择 **New**。
New File 对话框显示。
2. 输入该事件的名称：**braking_in_a_turn**。这是你要存储在硬盘或网络驱动器上文件的名称，并被自动赋予 .xml 的后缀。
3. 选择 **OK**。

MSC.ADAMS 将生成文件并定义第一个基本操控 MINI_1，参数如下：


- ◆ Active: yes
- ◆ Abort Time: 10.0
- ◆ Step Size: 0.01
- ◆ H-Max: 0.01

练习 15 整车装配

■ 生成三个基本操控

- ◆ 要完成 braking-in-turn 事件，你必须生成三个基本操控过程。

■ 要生成三个基本操控：

1. 要进入生成基本操控界面，选择 。
生成基本操控界面出现。注意，缺省的，第一个基本操控过程 MINI_1，已经被定义了。
2. 要生成更多的基本操控过程，在 **Name** 一栏内输入 **MINI_2**, 并选择 **Add**。
3. 要生成最后一个基本操控过程，在 **Name** 一栏内输入 **MINI_3**, 并选择 **Add**。

基本操控窗口显示如下：

Name	Active	Abort Time	Step Size	H-Max
MINI_1	yes	10.0	0.01	0.01
MINI_2	yes	10.0	0.01	0.01
MINI_3	yes	10.0	0.01	0.01

练习 15 整车装配

- ◆ 你可以指定换挡的参数，但是，本练习中，按照缺省的设置就足够了。你也可以同初始的挡位、速度一样，指定在第一步基本操控中静平衡求解的设置，如下所述。
- ◆ 由于你所指定的第一步基本操控过程为直线加速，你将使用直线静平衡求解的设置。有关在驾驶机器中所有的静平衡设置方法，参见 ADAMS/Car 在线帮助手册中 Analyze 部分。
- 4. 要修改直线段静平衡求解的设置，选择 **Static Set-up** 再选择设置 **Task** 为 **straight**。注意其余的参数并未使用，因为这些参数将在准静态求解的过程中被计算的初始值所覆盖。
你必须设置车辆的初始速度。所设置的初始速度的值与使用 Event Builder 中所设的单位有关。

练习 15 整车装配

5. 要检查单位或进行修改，从菜单 **Settings** 下选择 **Units**。
缺省的，长度单位设置为 meters 而时间单位设置为 seconds。因此，速度的单位为 meters/second。
6. 要验证速度的单位是否为 meters/second，在 Event Builder 对话框口顶部的 **Speed** 文本栏内点击鼠标左键，然后看看在 **Current Field Unit** 文本栏内所显示的内容。
7. 要设置目标车辆速度，在 **Speed** 文本栏内输入 **27.7** (如前所述，单位为 meters/second)，该值大约对应 100 km/h。
8. 要设置初始档位，在 **Gear** 文本栏内输入 **5**。
9. 在 **MINI_2** 的 **Abort Time** 内修改结束时间为 **5**。
10. 要存储基本操控设置，选择 **Save**。

MSC.ADAMS 存储文件 braking_in_a_turn.xml 在当前工作目录下。

练习 15 整车装配

■ 定义基本操控的参数

- ◆ 现在你需要定义三个基本操控过程的参数。你需要定义每个基本操控过程的应用和结束条件。
- ◆ 应用包括转向(steering)、油门(throttle)、刹车(brake)、离合(clutch)和档位(gear)。对每个应用，你需要定义驾驶机器(Driving Machine)的控制方式 (开环-open loop，闭环-closed loop，或人工-human)。
- ◆ 结束条件定义车辆的状态，达到某个状态即结束该基本操控步骤。结束条件的定义有相当的弹性，可以用于多个目的。下面为一些结束条件的例子：
 - 结束条件可以只有一个条件，比如时间 time 大于等于 5 seconds。
 - 结束条件可以只有一个在某个时间周期内在一定的误差范围内的值。比如横向加速度为 0.5 g，超过 2 秒钟，误差为 0.05 g。

练习 15 整车装配

- 多个结束条件，比如时间超过 5 秒钟或车辆速度低于 10 km/h。
- 组合结束条件，比如车辆速度等于 100 km/h (在一定的误差范围内) 并且纵向加速度等于 0 g (在一定的误差范围内)。还允许结束条件组，允许多个结束条件满足时，某一个基本操控过程终止。
- ◆ 更多有关结束条件，参见 ADAMS/Car 的在线帮助手册 Analyze 部分。

练习 15 整车装配

■ 要定义基本操控过程 MINI_1 的参数：

1. 要打开基本操控过程编辑器，双击你要编辑的基本操控过程的名字。此处为 **MINI_1**。
2. 对第一个基本操控过程，车辆应该是直线行驶并保持恒定的速度。要达到这一目标，设置下列参数：

◆ **Steering** 部分

- ◆ **Actuator Type:** rotation
- ◆ **Control Method:** machine
- ◆ **Control Mode:** Absolute
- ◆ **Steer Control:** straight

转向控制设置使驾驶机器驱动汽车保持直线行驶。

练习 15 整车装配

◆ Throttle 部分

- ◆ **Control Method:** machine
- ◆ **Control Mode:** Absolute
- ◆ **Speed Control:** maintain
- ◆ **Velocity:** 27.7

速度设置使得驾驶机器(Driving Machine)能够保持前面定义的稳态车速。油门将控制并保持这一车速，类似于巡航控制系统。

● Braking 部分

- **Control Method:** machine

油门和刹车设置控制车辆的纵向速度。

● Gear 部分

- **Control Method:** open
- **Control Type:** constant
- **Control Value:** 5

保持同样的档位。

● Clutch 部分


- **Control Method:** open
- **Control Type:** constant
- **Control Value:** 0.0

练习 15 整车装配

- ◆ 对第一个基本操控过程，你使用驾驶机器(Driving Machine) 的转向和油门控制，使得车辆能够在直线段行驶过程中保持恒速。
- ◆ 第一个基本操控过程的目标是达到动态稳定的结束条件。虽然对这类事件而言不是绝对必须的，但它有助于说明驾驶机器中一些重要的使用方式。要达到稳定的状态，车辆必须经过一段直线段(非常低的横向加速度)，达到稳定速度 (非常低的纵向加速度)。你可以使用结束条件将此两个条件形成一个组，以满足稳态的要求。

练习 15 整车装配

■ 要生成 MINI_1 的结束条件：

1. 选择 **End Conditions**。
2. 选择 。
3. 在 **Name** 文本栏内输入 **END_2**。
4. 选择 **Add**。
5. 修改结束条件如下：

◆ END_1

◆ Condition Type:	lon_accel
◆ Test Type:	==
◆ Trigger Value:	0.0
◆ Error:	0.05
◆ Filter Time:	2.0
◆ Delay Time:	0.0
◆ Group Name:	MINI_1

练习 15 整车装配

◆ END_2


- ◆ Condition Type: lat_accel
- ◆ Test Type: ==
- ◆ Trigger Value: 0.0
- ◆ Error: 0.05
- ◆ Filter Time: 2.0
- ◆ Delay Time: 0.0
- ◆ Group Name: MINI_1

注意使用 MINI_1 作为组的名字。因为 END_1 和 END_2 共有一个名字，对于该基本操控步骤来说，需要两个结束条件同时满足才可。

6. 选择 **Save**。

练习 15 整车装配

■ 要定义基本操控过程 MINI_2 的参数：

1. 选择 .
2. 在基本操控过程的名字 **MINI_2** 双击。
3. 设置参数如下：

◆ **Steering** 部分

- ◆ **Actuator:** rotation
- ◆ **Control Method:** machine
- ◆ **Control Mode:** Absolute
- ◆ **Steer Control:** skidpad
- ◆ **Entry Distance:** 10.0
- ◆ **Radius:** 120
- ◆ **Turn Direction:** Right

这些参数将使车辆在第二步操控过程的开始阶段右转。

练习 15 整车装配

- **Throttle部分**

- **Control Method:** machine
- **Control Mode:** Absolute
- **Speed Control:** maintain
- **Velocity:** 27.7

这些参数使驾驶机器保持前面 MINI_1 直线段的速度。转向参数不做修改。

- **Braking部分**

- **Control Method:** machine

- **Gear部分**

- **Control Method:** open
- **Control Type:** constant
- **Control Value:** 5

保持同样的档位不变。

练习 15 整车装配

- **Clutch部分**
 - **Control Method:** open
 - **Control Type:** constant
 - **Control Value:** 0.0

在第二步的基本操控步中，你设置驾驶机器的转向和油门，使车辆右转并控制转向半径为 120 m。

下一步，你将生成两个结束条件以保证达到稳态的转向条件。

练习 15 整车装配

■ 要生成 MINI_2 的结束条件：

1. 选择 **End Conditions** 键。
2. 在 **Name** 文本栏内输入 **END_1**。
3. 选择 **Add**。
4. 同第一个一样，生成第二个结束条件 **END_2**。
5. 修改结束条件如下：

◆ **END_1**

◆ Condition Type:	lon_accel
◆ Test Type:	==
◆ Trigger Value:	0.0
◆ Error:	0.05
◆ Filter Time:	2.0
◆ Delay Time:	0.0
◆ Group Name:	MINI_2

练习 15 整车装配

- ◆ **END_2**
 - ◆ **Condition Type:** curvature
 - ◆ **Test Type:** ==
 - ◆ **Trigger Value:** 0.00833
 - ◆ **Error:** 0.00005
 - ◆ **Filter Time:** 2.0
 - ◆ **Delay Time:** 0.0
 - ◆ **Group Name:** MINI_2

这些结束条件保证转向半径近似等于 120 m , 速度近似等于 100 km/h。

6. 选择 **Save**。

练习 15 整车装配

■ 要定义基本操控步骤 MINI_3 的参数：

1. 选择 
2. 在基本操控步骤 MINI_3 的名字上双击鼠标左键。
3. 设置基本操控步骤 MINI_3 的参数如下：

◆ Steering 部分

- ◆ Actuator: rotation
- ◆ Control Method: machine
- ◆ Control Mode: Absolute
- ◆ Steer Control: skidpad
- ◆ Entry Distance: 0.0
- ◆ Radius: 120
- ◆ Turn Direction: Right

这些参数保持前面操控步骤的转向半径不变。

◆ Throttle部分

- ◆ Control Method: machine

练习 15 整车装配

- **Braking**部分

- **Control Method:** machine
- **Speed Control:** lon_accel
- **Start Time:** 1.0
- **Long. Acc.:** - 3.0

这些参数在第三步操控步骤的开始阶段将油门降为 0，并控制车辆的加速度为 -3.0 m/s²。

- **Gear**部分

- **Control Method:** machine
由驾驶机器控制档位的选择。

- **Clutch**部分

- **Control Method:** machine
由驾驶机器控制离合器的选择。

在第三步基本操控步骤中，使用驾驶机器控制车辆的转向半径不变，同时刹车，加速度为 3 m/s²。你将使用单个的结束条件，就是速度。你将触发仿真结束条件，当车辆的速度低于 10 km/h，相当于 2.77 m/s 时。

练习 15 整车装配

■ 要生成 MINI_3 的结束条件：

1. 选择 **End Conditions**。
2. 在 **Name** 文本栏内，输入 **END_1**。
3. 选择 **Add**。
4. 修改结束条件如下：

◆ **END_1**

◆ Condition Type:	velocity
◆ Test Type:	<<
◆ Trigger Value:	2.77
◆ Error:	0.01
◆ Filter Time:	0.0
◆ Delay Time:	0.0

练习 15 整车装配

5. 选择 **Save**。

缺省的，MSC.ADAMS 存储操控文件在当前工作目录下。如果你不确定当前工作目录的位置。可以：

- ◆ 在 ADAMS/Car 下，从菜单 **File** 下选择 **Select Directory**。
你会发现该目录的位置，类似于 C:\Documents and Settings\
user_name 或 /home/usr/username。

练习 15 整车装配

■ 运行分析

要运行分析：

1. 在 ADAMS/Car 中：

- ◆ 从菜单 **Simulate** 下选择 **Full-Vehicle Analysis**，再选择 **Driving Machine Control File (DCF) Driven**。
- ◆ 在 **Driver Control Files** 文本栏内输入你在本练习中所生成的新事件的名字 (**braking_in_a_turn.xml**)。
- ◆ 填入其它参数。有关如何填入其它参数，可以按 **F1** 键。
- ◆ 现在你就可以使用 Event Builder 运行整车的仿真了。

附录 A 分析的例子

- 本附录介绍一些你如何使用 ADAMS/Car 分析你的车辆的一些概念性的东西。

分析的例子

■ 本附录内容：

- ◆ 分析类型
- ◆ 搜集有关模型的数据
- ◆ 悬架的包络空间分析
- ◆ 悬架的运动学分析
- ◆ 悬架的柔顺性分析
- ◆ 静载耐久性分析
- ◆ 动载耐久性分析
- ◆ 前悬架分析
- ◆ 整车的设计和分析

分析的类型

- 下面列出在 ADAMS/Car 可以做的分析
 - ◆ 悬架的包络空间分析
 - ◆ 悬架的运动学分析
 - ◆ 悬架的柔顺性分析
 - ◆ 静载耐久性分析
 - ◆ 动载耐久性分析
 - ◆ 前悬架分析
 - ◆ 整车的设计和分析
- 在后面的篇幅，给出进行这些分析的一些指导。此指导只是列出 ADAMS/Car 中提供的部分分析功能，另外，要进行以上列出的分析，可能需要其它的一些模块 (比如：ADAMS/Durability)。



更多有关特殊的分析功能，参见 ADAMS/Car 在线帮助。

搜集有关模型的数据

- 搜集必要的数​​据，建立你要分析的悬架的MSC.ADAMS模型。在你从各种渠道搜集到相关的数​​据 (CAD 几何外形、来自内部试验厂或供应商的衬套数​​据、减震器数​​据等等)后，你可以建立 ADAMS/Car 的模型了。

悬架的包络空间分析

- ◆ 你一旦生成 MSC.ADAMS 概念悬架的模型后，你可以将你的虚拟悬架置于虚拟试验台架上 (ADAMS/Car 的标准部分) 并运行一系列的仿真过程以检验包络空间和动态干涉方面的问题。此项分析的目的就是检查一下在轮胎弹跳和滚转过程中部件不会发生碰撞。
- ◆ 同样地，如果各部件的几何外形知道的话，你可以证实轮胎和车轮有足够的间隙，符合标准的要求。这个阶段的分析的目的是在 ADAMS/Car 的环境下，粗略地检查一下部件之间的干涉情况。
- ◆ 进一步的分析，可能是将 MSC.ADAMS 在最大的弹跳/滚转情况下所得到的轮胎包络空间的分析结果带入你的 CAD 软件包中。你可以在 CAD 软件包中使用所得到的车轮包络空间的实体几何外形很容易的发现是否存在干涉问题，也能够直观的更好的显示出你的概念悬架中车轮所需要的总的空间范围。

悬架的运动学分析

- ◆ 解决了悬架的包络空间问题后，你就可以将虚拟的悬架置于虚拟的试验台架上并运行一系列的悬架运动学分析，以分析并理解悬架的运动学特性。两个分析有助于你了解悬架的运动学特性：一个是 parallel wheel travel () 以及另一个是 roll travel (车轮反向垂直跳动)。
- 1. 车轮同向垂直跳动分析(也称作 ride-travel event) 主要检查下列悬架矩阵：
 - ◆ Toe (also called bump steer)
 - ◆ Caster (后倾角)
 - ◆ Camber
 - ◆ Longitudinal recession (纵向轴距变化)
 - ◆ Lateral recession (横向轮距变化)
 - ◆ Wheel rate (vertical force versus amount of suspension vertical deflection)
- 2. 车轮反向垂直跳动分析主要检查下列悬架矩阵：
 - Roll steer (degrees of toe per degree of suspension roll)
 - Roll stiffness

悬架的运动学分析

- ◆ 悬架运动学分析的目的是调整悬架的几何尺寸以得到满意的运动学特性。如果出现了运动学方面的问题，可以提出设计的修改建议，改变悬架连接点的位置、控制臂 (control arms) 的长度和其它影响悬架运动学特性的一些几何参数。
- ◆ 同样地，悬架弹簧的特性也可以得到检查以保证车辆总体上性能的要求，包括悬架在车轮跳动和扭转过程 (ride and roll) 中对弹簧弹性的要求。使用附加的悬架组件，如抗扭转杆，也可以得到检查，包括抗扭转杆的几何尺寸等。

悬架的柔顺性分析

- ◆ 在分析了悬架的几何设计并得到了良好的运动学性能之后，你可以检查一下悬架的柔顺性能。虚拟的悬架模型将置于悬架试验台上并进行一系列的柔顺性能分析 (比如，静载荷分析)。通过这些分析可以得到下列矩阵：
 - Lateral force versus toe (for both parallel and opposing lateral force)
 - Lateral force versus camber (for both parallel and opposing lateral force)
 - Lateral force versus lateral displacement (for both parallel and opposing lateral force)
 - Longitudinal force versus toe (braking and acceleration forces)
 - Longitudinal force versus longitudinal displacement (braking and acceleration forces)
 - Aligning torque versus toe (parallel and opposing torques)
- ◆ 悬架的柔顺性分析的目的在于调整悬架衬套以得到合适的悬架柔顺性能。注意实际的连接铰链的刚度并非是无限制的，而且它们确实对悬架的性能有一定的影响。因此，在你的模型中考虑使用衬套代替理想的铰链连接应该讲是很好主意。

静载耐久性分析

- ◆ 悬架性能的下一步分析就是在 ADAMS/Car 中在车轮上施加静载工况并检查传递到悬架元件处的载荷 (悬架衬套、悬架弹簧等等)。这有助于更好地了解悬架的耐久性能。
- ◆ 典型情况下，进行静载分析是使用一系列悬架需要承受的最恶劣工况。这些载荷条件 (或工况) 采取多少个 g 的载荷形式表示。例如，悬架需要承受 3 g 的垂向载荷，2 g 的纵向载荷和 1 g 的横向载荷。这样的载荷条件通常称为 “321 g” 载荷工况。
- ◆ 使用 “g’s” 描述整车重量，并分配到前/后悬架上。在这种情况下，采用 “g’s” 形式表示的载荷可以用于不同的尺寸和重量的车辆的计算中。

静载耐久性分析

- ◆ 静载耐久性分析的目的是给出设计人员和有限元强度分析人员一个报告，报告中给出作用在悬架各部件(控制臂、衬套、弹簧等等)上所受到的最恶劣的载荷情况。这些数据随后可以传入到各部件(控制臂、衬套、弹簧等等)的有限元模型中进行强度方面的分析。一个结构静载荷报告通过载荷工况的后处理生成，从而得到悬架上部件所承受的总载荷。此报告可以传到有限元软件 (NASTRAN、ANSYS、ABAQUS等等)进行强度分析。
- ◆ 对于设计工程师来说，很少能够在项目的早期的这个阶段取得这些数据，因此静载耐久性分析可以提供在真实世界里首次洞察部件的载荷条件。这种方法是简单的粗略的，并不代表进行耐久性能分析最终的载荷分析结果，但在设计和分析的早期阶段，这些数据是相当有价值的。

动载耐久性分析

- 静载耐久性分析完成后，就可以进行更为直接的研究橡胶悬挂部件的性能。在此阶段，需要为悬架假定一个路面的截面形状 (例如，一个凹坑路面或测试得到的路面白噪声随机信号输入) 并利用 ADAMS/Car 中虚拟的四轮台柱振动分析工具。得到的MSC.ADAMS 动态载荷时间历程将被输入到 FEA 软件中进行最后的结构强度分析。

前悬架分析

- 除了上述列出的分析之外，对于前悬架而言，还要进行转向系统特性的分析检查，以了解总的转向传动比，转向线性特性，Ackerman转向特性和转向柱管万向节相位滞后(保证转向过程不出现滞重感)。

整车的设计和分析

- 一旦在 ADAMS/Car 中后悬架、前悬架和转向系统都进行了模拟，功能化数字车辆分析的下一步流程就是要进行整车性能的分析了。因为各个不同的子系统已经建立了，组装成 ADAMS/Car 中的整车装配就是一件相当简单的事情了。
- ADAMS/Car 的模型将被驱动以进行一系列的虚拟仿真过程，以理解整车的性能。整车可以进行的分析包括：
 - ◆ Static vehicle trim analysis to trim the vehicle to a particular ride height —— 车辆静载调整分析，车辆调整到特定的走行高度分析。

整车的设计和分析

- ◆ Double lane change (Moose Test) to examine rollover propensity in an accident-avoidance maneuver——双移线分析（驼鹿试验），检验车辆在紧急避让操纵性能分析中的倾覆特性
- ◆ Constant radius cornering to examine understeer/oversteer behavior of the vehicle and generate an understeer budget report ——等半径转向，检验车辆的不足转向/过转向特性，并生成不足转向预算报告
- ◆ Brake drift to examine amount of lateral drift during a moderate-braking maneuver——制动漂移，检验在适当的制动操作过程中车辆的横向漂移量
- ◆ Brake in a turn to examine dynamic braking and turning stability issues ——转向过程中制动，检验动态制动特性和转向稳定性方面问题
- ◆ Step steer to examine lateral acceleration and yaw rate overshoot during highly dynamic maneuver——阶跃转向，以检验在高度动态操纵过程中横向加速度和侧摆率的超调量
- ◆ Frequency response to examine vehicle dynamics metrics in the frequency domain——频响特性，以检验车辆在频域的动态矩阵
- ◆ Dynamic durability to determine the behavior of the vehicle as it drives over potholes, bumps, and other three dimensional road obstacles——动态耐久性能，以检验车辆通过坑洼、颠簸等三维路面障碍时的性能。

附录 B 四轮台柱式垂向激励试验台

- 本练习将讲述如何建立 ADAMS/Car 用户化的试验台，用户自定义仿真过程以及如何生成用户个人的 ADAMS/Car 二进制启动文件。
- 你将从由 ADAMS/Car 模板建模工具生成的现成的试验台开始进行练习。该练习将回顾建立试验台的内部工作过程，包含如何建立试验台的每一个步骤，并详细介绍如何编辑并生成包含该试验台在内的用户化 ADAMS/Car 的二进制文件。

四轮台柱式垂向激励试验台

- 完成本练习，你必须用到 7 个相关文件，包括四轮台柱式垂向激励试验台的模型文件、宏文件和界面相关文件：

- ◆ acar_build.cmd
- ◆ acme_3PostRig.cmd
- ◆ acme_four_sim.cmd
- ◆ macros_ana.cmd
- ◆ mac_ana_ful_fou_sub.cmd
- ◆ acme_4PostRig.cmd
- ◆ fourpost_header.txt

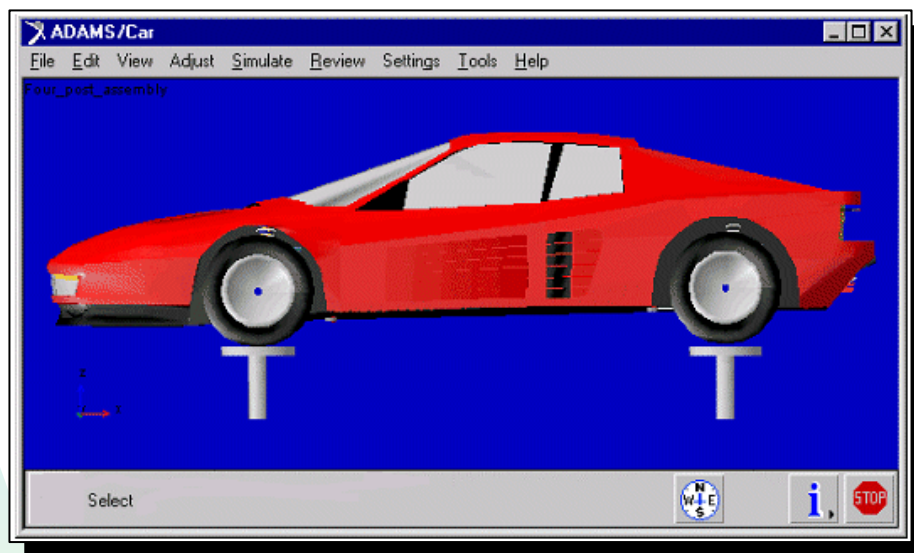
从你的指导老师那里得到这些文件。

四轮台柱式垂向激励试验台

■ 本附录内容：

- ◆ 介绍
- ◆ 在 ADAMS/Car 中生成试验台模板
- ◆ 生成个人二进制文件的准备工作
- ◆ 在 UNIX 系统上
- ◆ 在 ADAMS/Car 中增加用户化的分析过程

四轮台柱式垂向激励试验台



介绍

- ◆ 本练习的目标是检查整车及其悬架的垂向动力学性能。试验的结果可以在频域进行后处理，以研究车辆在各种行驶状态下的系统振动频率及其相应的振动水平。也可以对车辆的垂向动力学性能对操纵性能的影响得到附带的深入了解，包括下面的一些：

四轮台柱式垂向激励试验台

- Front to rear modal balance
 - Suspension to body transfer function gain and phase
 - Suspension to tire transfer function gain and phase
 - Tire contact patch vertical load variation
- ◆ 试验过程通过将整车模型装配到一个专门的四轮台柱式试验台上完成。该试验台只是简单地定义四个可动部件代表支撑车辆的四个轮胎的台柱。这些轮胎的支撑台柱被约束，只允许沿着垂直方向移动，同时，这个垂直方向的移动由位移激励 (驱动) 控制。支撑台柱和车辆轮胎之间唯一的连接关系就是轮胎本身的摩擦力。因为 Delft 类型的轮胎支持 “0” 速度情况下轮胎的摩擦力的计算，这些是进行车辆垂向动力学分析需要加在车辆上的约束条件。

四轮台柱式垂向激励试验台

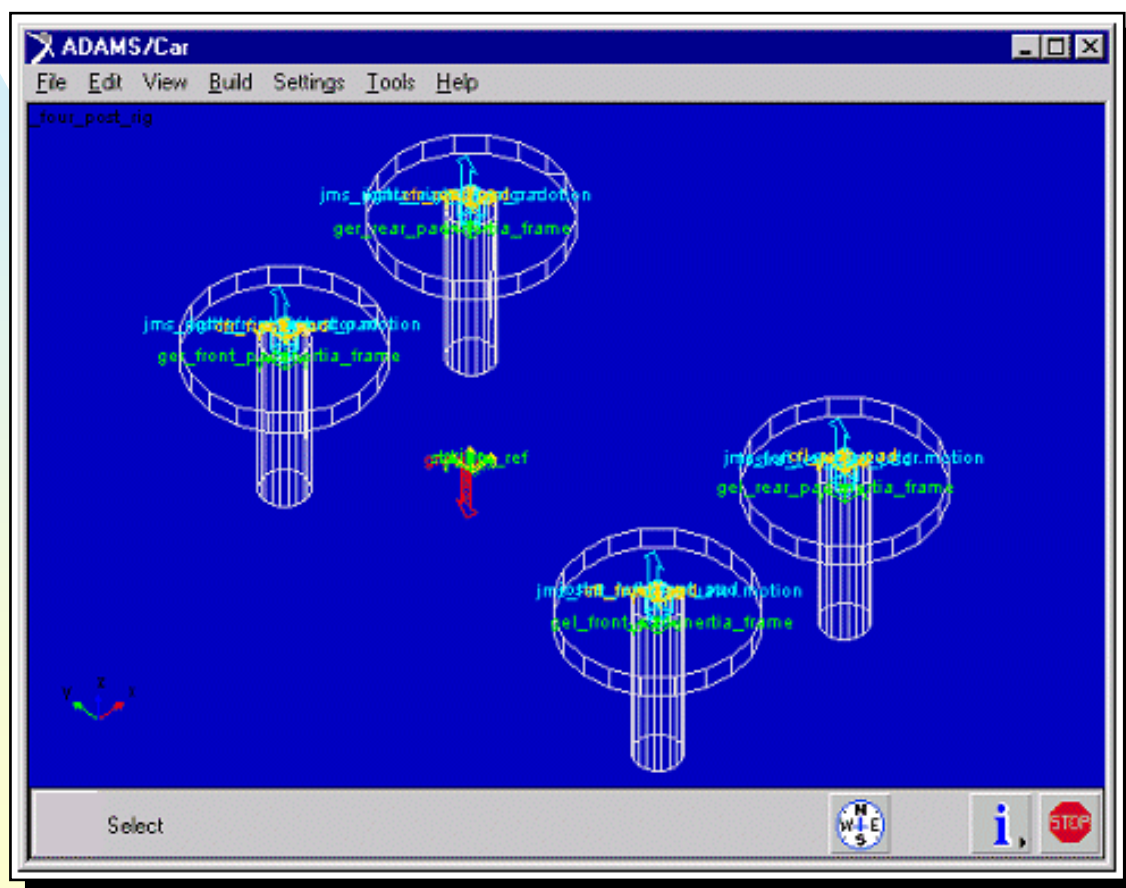
- ◆ 每个垂向的激励分别由一个分析函数所确定。分析函数用来描述激励器的位移在时域的曲线并局限于等幅扫频正弦输入信号激励，频率在预定的频率范围内。使用分析函数控制四个台柱的运动有下面几种组合方式：
 - Heave: all tire pads move vertically in phase.
 - Pitch: the front tire pads move 180 degrees out of phase with the rear tire pads.
 - Roll: the left tire pads move 180 degrees out of phase with the right tire pads.
 - Warp: the left-front and right-rear tire pads move 180 degrees out of phase with the right-front and left-rear pads.

四轮台柱式垂向激励试验台

■ 试验台描述

- ◆ 四轮台柱试验台是在 ADAMS/Car 的模板建模模式下生成的，其名字为 __acme_4PostRig.tpl。下面是其工作机理的简单描述以及其是如何与标准的 ADAMS/Car 仿真类型相关联。四轮台柱试验台的主要角色 (major role) 是 analysis，并包含四个一般部件 (general parts)。在每一个台柱上定义一个垂直方向的移动副，每个移动副上对应一个激励。所有的台柱及其约束、激励的位置等等都被参数化定义在大地平面 (X 和 Y)，由来自于悬架系统中轮胎中心位置的信息交换器所确定。其垂向位置由 std_tire_ref 标记点的 Z 值确定。std_tire_ref 标记点的 Z 高度在装配过程中自动地确定，这样的话，它可以表示车辆轮胎接触的平均高度。

四轮台柱式垂向激励试验台



四轮台柱式垂向激励试验台

- 四轮台柱试验台需要的的输入：仿真过程的描述
 - ◆ 分析过程的输入参数分成两组：一组参数是对于所有的分析过程，而另一组则包含专门对四轮台柱试验台的输入。四轮台柱试验台需要的输入由用户控制，以定义进行垂直方向激励需要的边界条件。
 - ◆ 始终需要的输入：
 - Output Prefix
 - End Time
 - Number of Steps
 - Type of Analysis (Interactive, Background)
 - Analysis Log File (Yes, No)
 - ◆ 四轮台柱试验台需要的输入：
 - Peak Displacement
 - Displacement Units (m, mm, inch, and so on)
 - Frequency Range (Units hardcoded to Hz)
 - Excitation Mode (Heave, Pitch, Roll, Warp)

四轮台柱式垂向激励试验台

■ 装配和仿真过程需要的输入

- ◆ 下列步骤讲述在提交了仿真之后试验台的内部所进行的工作。车辆和试验台之间的装配过程在不考虑前面章节中所描述的用户输入的参数的话是相似的。然而，装配模型的修改将依这些参数的变化而改变。
- ◆ 装配过程：
 1. 一些子系统级别的检查，如同在整车操纵分析的装配过程一样，需要进行额外的检查，以保证整个装配中不会多于四个车轮和两个悬架。
 2. 装配车辆和试验台。位置信息交换器将定位台柱在大地的 X – Y 平面上。
 3. 考虑轮胎并重置 GFORCE 的参考标记点到相应的台柱上。例如，左前部的轮胎力的参考标记点需要属于试验台的一般部件 gel_front_pad 上。

四轮台柱式垂向激励试验台

4. 基于所有车轮平均的接触位置分配 std_tire_ref 标记点的 Z 坐标 (与在整车仿真中的概念一样)。
 5. 设置轮胎内置路面的特性文件为 mdi_2d_flat.rdf , 并且不需要生成任何的路面几何外形。
 6. 依据用户的输入数据修改激励的 (jmf_left_front_actuator、jmf_right_front_actuator等等)的分析函数。
- ◆ 下列函数需要基于分析驱动过程的信号分配给相应的激励台柱：

```
Left Front = LF_phase*Peak_Amplitude*sin(1/2*360D*Freq_Range/End_Time*Time**2)
Right Front = RF_phase*Peak_Amplitude*sin(1/2*360D*Freq_Range/End_Time*Time**2)
Left Rear = LR_phase*Peak_Amplitude*sin(1/2*360D*Freq_Range/End_Time*Time**2)
Right Rear = RR_phase*Peak_Amplitude*sin(1/2*360D*Freq_Range/End_Time*Time**2)
```

四轮台柱式垂向激励试验台

◆ 赋予相位变量下列数值：

- Heave Mode: LF_Phase, RF_Phase, LR_Phase, RR_Phase = 1.0
- Pitch Mode: LF_Phase, RF_Phase = 1.0 & LR_Phase, RR_Phase = -1.0
- Roll Mode: LF_Phase, LR_Phase = 1.0 & RF_Phase, RR_Phase = -1.0
- Warp Mode: LF_Phase, RR_Phase = 1.0 & RF_Phase, LR_Phase = -1.0

◆ 试验台然后进行下列的仿真过程：

- 类似于整车的分析过程，需要提交仿真给 solver。仿真需要一个静配合分析，初始速度为 0.0，然后进行结束时间为用户指定的时间的动力学仿真。积分器中最大的步长 (hmax) 至少应该设置为最大频率的 1/10。例如：用户设置的最大频率为 20Hz 的话，那么 hmax 应该为 $1/10 \times 1/20 = 1/200$ (0.005)。这样可以在仿真过程中避免丢掉一些感兴趣的信号。

四轮台柱式垂向激励试验台

- 在 ADAMS/Car 中生成试验台的模板
 - ◆ 在 ADAMS/Car 中要完成增加试验台的任务最基本的条件是对 ADAMS/Car 模板建模工具具有一个通透的理解。用户如果想要在 ADAMS/Car 中建立试验台，非常重要的一点就是要非常了解信息交换器和激励的概念。你对本章的练习逐步地进行时，你应该参考 ADAMS/Car 在线帮助中模板部分。参考包含在 ADAMS/Car 中标准的试验台模板也是非常有用的。
 - ◆ 在 ADAMS/Car 中的试验台与 ADAMS/Car 中的模板几乎是可以完全相比拟的。试验台与其它模板的最主要的区别就在于试验台中包含激励元件如驱动或者力，用来激励整个装配系统。

四轮台柱式垂向激励试验台

- ◆ 在 ADAMS/Car 中生成试验台模板的过程与生成一个标准的模板几乎是完全一样的，只有少数的差别。建立试验台模板的步骤如下：
 - 生成试验台模板
 - 存储试验台模板
 - 修改试验台模板

■ 开始练习

- ◆ 试验台模板是在 ADAMS/Car 的模板建模模式下进行的。类似于一个常规的模板，一个试验台的模板能够包含通过约束或载荷连接在一起的部件。与常规的模板类似，试验台模板同样还要包含 信息交换器，以使试验台在装配时能够与其它模板交换一些各自需要的信息。与常规的模板不同之处在于试验台模板同样还要包含 驱动以激励系统。

四轮台柱式垂向激励试验台

- ◆ 由于模板和试验台是如此的相似，全面完整的讲述如何生成试验台似乎显得有些多余。取而代之的办法是，对于有关建立模板和信息交换器的一些信息，可以 ADAMS/Car 在线帮助中模板部分。
- ◆ ADAMS/Car 处理试验台的方式就像模板。然而，为了能够在装配中使用试验台，需要将试验台模板转化为试验台模型文件 (.cmd) 并且需要生成一个个人的 ADAMS/Car 二进制文件 (.bin)。当然了，你能够在 ADAMS/Car 的界面下很容易地生成一个新的试验台模板，但通常的做法是利用已有的模板，以更好的理解 MSC.ADAMS的功能。
- ◆ 从在模板建模模式下修改现有的试验台模型文件 (.cmd) 开始。先找到文件 acme_3PostRig.cmd。此文件为试验台的模型文件，此试验台中包含了三个起作用的车轮台柱。

四轮台柱式垂向激励试验台

- 加载试验台模型文件 (.cmd) 到 ADAMS/Car 模板界面
 - ◆ 如果你想打开现有的试验台模型文件 (.cmd) 并使用 ADAMS/Car 的模板界面进行编辑，按照下列步骤进行。
 - ◆ 注意：如果你在编辑试验台模板时，使用个人 (private) 或定点 (site) 的 ADAMS/Car 二进制方式需要小心一些。如果你已经建立了一个试验台，并且个人的二进制文件正在使用，你不能以同样的名字对试验台进行编辑。

四轮台柱式垂向激励试验台

要加载试验台模型文件：

1. 将试验台模型文件 **acme_3postrig.cmd** 移至你的个人的 car 数据库的数据表目录下 (比如, C:\private.cdb\templates.tbl\).
2. 将文件 **acme_3postrig.cmd** 改名为 **_acme_4PostRig.tpl**。确认模板的名字前面有两个下划线 (__)。
3. 在文本编辑器内打开此文件。
4. 插入下面的头文字：
 - ◆ 注意 TEMPLATE_NAME 必须与文件名字完全匹配 (不包括扩展名)。

```
$-----MDI_HEADER  
[MDI_HEADER]  
FILE_TYPE = 'tpl'  
FILE_VERSION = 13.7  
FILE_FORMAT = 'ASCII'  
HEADER_SIZE = 9  
(COMMENTS)  
{comment_string}  
'ADAMS/Car analysis template'
```

四轮台柱式垂向激励试验台

```
$-----TEMPLATE_HEADER  
[TEMPLATE_HEADER]  
TEMPLATE_NAME = '__acme_4PostRig'  
MAJOR_ROLE = 'analysis'  
TIMESTAMP = '2003/10/02,10:00:23'  
HEADER_SIZE = 6
```

- **提示：**你可以从文件 four_post_header.txt 中拷贝相应部分的文字到该文件中，该文件在 four_post_test_rig_files 中。
5. 在文件 acme_4postrig.cmd 有一个变量称作 model_class。搜索 model_class。改变此变量的赋值如下：(此变量必须设置为 template，因为模板建模器需要使用它)。

```
variable create &  
  variable_name    = __acme_4PostRig.model_class &  
  string_value     = "template" &  
  comments         = "Memory for ADAMS/Car model class"  
!
```

6. 存储此文件。

四轮台柱式垂向激励试验台

- 在 ADAMS/Car 模板建模界面下打开此模板
 - ◆ 你刚才使用的文件为一个 ADAMS/Car 的模板，包含一个试验台的模型，模型中只有三个起作用的台柱，本节中，你将激活第四个(左-前部)的台柱。
- 要打开试验台：
 1. 确认你的个人 **.acar.cfg** 文件中指定你为一个专家用户，这样你可以使用 ADAMS/Car 模板建模界面。
 2. 启动 ADAMS/Car 并在提示窗口中切换到 **template-builder** 模式下。
 3. 从菜单 **File** 下选择 **Open**。
 4. 在 Template Name 文本栏内点击鼠标右键并选择 **Search**，然后再选择 **<private>/ templates.tbl**。
 5. 选择 **__acme_4PostRig.tpl** 模板文件，然后选择两次 **OK**。
 6. 确认图标处于开启状态，并设置视图到 **Front Iso** 并设置视图为全显示状态。

四轮台柱式垂向激励试验台

- 在 ADAMS/Car 模板建模界面下修改试验台模板
 - ◆ 要激活激励，你可以在试验台中增加一系列的对象：一个止档，一个反向止档，一个约束连接，一个约束运动驱动和一个约束力驱动。
- 要增加止档：
 1. 从菜单 **Build** 下选择 **Forces**，再选择 **Bumpstop**，然后再选择 **New**。
 2. 在 **Bumpstop Name** 文本栏内输入 **front_extension_stop**。
 3. 指定 **I Part** 为 **gel_front_pad**，可以直接输入，又可以通过在文本栏内点击鼠标右键选择 Part,再选择 **Pick**, 然后再选择部件 **gel_front_pad** 的方法完成。
 4. 指定 **J Part** 为 **ground**，可以通过在文本栏内点击鼠标右键选择 **Guesses** 再选择部件 **ground**的方法完成。

四轮台柱式垂向激励试验台

5. 指定 **I Coordinate Reference** 为 **cfl_Front_Actuator_Base**。
6. 指定 **J Coordinate Reference** 为 **cfl_Front_Pad**。
7. 在 **Property File** 文本栏内选择
<shared>\bumpstops.tb\mdi_0001.bum 作为特性文件。
8. 在 **Clearance**文本栏内输入 **127**。
9. 选择 **OK**。

注意：由于几何对称性的关系，不仅仅是左前侧的止档生成了，同时右前侧三止档也同时生成了。

四轮台柱式垂向激励试验台

■ 要生成反向止档：

1. 从菜单 **Build** 下选择 **Forces**，再选择 **Reboundstop**，然后再选择 **New**。
2. 在 **Reboundstop Name** 文本栏内输入 **front_retract_stop**，这样做，主要是保证名字的一致性。
3. 设置 **I Part** 为 **gel_front_pad**。
4. 设置 **J Part** 为 **ground**。
5. 设置 **I Coordinate Reference** 为 **cfl_Front_Actuator_Base**。
6. 设置 **J Coordinate Reference** 为 **cfl_front_pad**。
7. 在 **Property File** 栏内输入
<shared>\reboundstops.tbl\mdi_0001.reb。
8. 在 **Clearance** 栏内输入 **127**。
9. 选择 **OK**。

注意：由于几何对称性的关系，不仅仅是左前侧的反向止档生成了，同时右前侧的反向止档也同时生成了。

四轮台柱式垂向激励试验台

- 要在激励台柱和激励台柱支座之间增加一个约束连接：
 1. 从菜单 **Build** 下选择 **Attachments**再选择 **Joint**然后再选择**New**。
 2. 指定 **Joint Name** 为 **left_front_pad**。
 3. 指定 **I Part** 为 **ground**。
 4. 指定 **J Part** 为 **gel_front_pad**。
注意：I 和 J 部件的顺序，在决定下一步中约束运动驱动的方向是相当重要的。
 5. 设置 **Type** 为 **single**。
 6. 设置 **Joint Type** 为 **translational**。

四轮台柱式垂向激励试验台

7. 设置 **Location Dependency** 为 **Delta location from coordinate**。
8. 设置 **Location Coordinate Reference** 为 **cfl_front_pad**。
9. 设置 **Location** 为 **0,0,0** 及 **local**。
10. 设置 **Orientation Dependency** 为 **Delta orientation from coordinate**。
11. 设置 **Construction Frame** 为 **cfl_front_pad**。
12. 设置 **Orientation** 为 **0,0,0**。
13. 选择 **OK**。

四轮台柱式垂向激励试验台

■ 要建立约束运动驱动：

1. 从菜单 **Build** 下选择 **Actuators**，再选择 **Joint Motion**，再选择 **New**。
2. 指定 **Actuator Name** 为 **left_front_actuator** 以保证名字一致。
3. 通过选择你刚刚生成的约束，指定 **joint** 为 **jostra_left_front_pad**。
4. 在 **Application** 栏内输入 **pad_excitation**。
5. 在 **Identifier** 栏内输入 **left_front_actuator**。
6. 在 **Function** 栏内输入 **15*sin(720d*time)**。此处只是简单的填一下而已，将来在进行分析的时候，分析的宏将会覆盖它。
7. 指定 **Time Derivative** 为 **displacement**。

四轮台柱式垂向激励试验台

8. 设置 **Force Limits** 为 $-5e4, 5e4$ 。
9. 设置 **Displacement Limits** 为 $-1000, 1000$ 。
10. 设置 **Velocity Limits** 为 $-1e4, 1e4$ 。
11. 设置 **Acceleration Limits** 为 $-9.8e4, 9.8e4$ 。
12. 在 **Units** 栏内输入 mm。
13. 选择 **OK**。

四轮台柱式垂向激励试验台

■ 要建立约束力激励：

1. 从菜单 **Build** 下选择 **Actuators**，再选择 **Joint Force**，再选择 **New**。
2. 指定 **Actuator Name** 为 **If_force_actuator**。
3. 选择 **Joint** 为 **jostra_left_front_pad**。
4. 在 **Application** 栏内输入 **Left front actuator force**。
5. 在 **Identifier** 栏内输入 **Left front actuator force**。
6. 在 **Function** 栏内点击鼠标右键并选择 **Function Builder**。
7. 粘贴下面的函数到文本栏内：

```
(STEP(TIME,0,1,0.002,0))*  
(1e5)*VARVAL(if_actuator_disp))  
+STEP(VARVAL(if_actuator_vel),pvs_Friction_Saturation_Velocity,  
-pvs_Friction_Saturation_Force,pvs_Friction_Saturation_Velocity  
,pvs_Friction_Saturation_Force)  
-VARVAL(if_actuator_force)
```

8. 选择 **OK**。

四轮台柱式垂向激励试验台

■ 存储试验台模板

- ◆ 试验台模板需要被存储到一个文件中，就像常规的模板一样。存储的格式应该是 ASCII 格式，以方便修改。修改是必要的，后面的章节将介绍该部分内容。

■ 要存储模板：

1. 从菜单 **File** 下选择 **Save As**。
2. 确认文件的格式设置为 **ASCII**，并且 **Zero Adams Ids** 被选了。
3. 选择 **Close Template**。
4. 选择 **OK**，出现提示时，选择不存储备份文件。

四轮台柱式垂向激励试验台

- 从 ADAMS/Car 的模板文件转换为试验台模型文件(.cmd)
 - ◆ 你必须手工修改试验台模板文件再将其转换为试验台模型文件 (.cmd)。
对 ADAMS/Car 中生成的 ASCII 格式的模板文件你必须作两处修改。

四轮台柱式垂向激励试验台

■ 要从一个模板文件生成试验台：

1. 在 ADAMS/Car 外部，从你的个人数据库目录 **templates.tbl** 下，拷贝文件 **__acme_4PostRig.tpl** 到你的个人目录 (C:\acar_private) 或其它目录下。
2. 将文件更名为 **acme_4PostRig.cmd**。
3. 使用文本编辑器，打开 **acme_4PostRig.cmd**。
4. 在文件 **__acme_4PostRig.tpl** 有一个名为 **model_class** 的变量，改变此变量的值，类似下面 (此变量需要设置为 test rig。):

```
variable create &  
  variable_name = __acme_4PostRig.model_class &  
  string_value = "testrig" &  
  comments = "Memory for ADAMS/Car model class"  
!
```

四轮台柱式垂向激励试验台

5. 删去在 ASCII 格式模板文件中起始部分的头信息。这是因为这些头信息在命令文件读入过程无法识别，并将输出错误信息。典型的头部信息参见前面 501 页的练习。
6. 删去从文件的起始行到 所有的行，包括 HEADER_SIZE 特征行在内。
7. 存储文件。

四轮台柱式垂向激励试验台

- 在试验台中所需要的 ADAMS/View 的变量
 - ◆ 在 ADAMS/Car 中的模板和试验台都需要一些 ADAMS/View 中的变量以确定该模板如何应用。所有的模板，包括试验台在内，都需要三个变量：major role、minor role 和 model class。试验台另外还有一个 ADAMS/View 的变量，名为 assembly class。
 - ◆ 在试验台中所需要的所有变量在下面进行介绍。前三个变量：role、minor_role 和 model_class 在试验台模板生成的时候就自动地生成了。你需要手工生成变量 testrig_class，这是只有试验台所需要的。

四轮台柱式垂向激励试验台

■ 主要角色 (Major role)

- ◆ ADAMS/Car 模板和试验台中的主要角色存储在 ADAMS/View 中的变量名称为 role。对于试验台而言，其主要角色始终是 analysis。在 ADAMS/Car 中生成试验台时，非常重要的一点就是保证正确设置其值。

```
variable create &  
variable_name = __acme_4PostRig.role &  
string_value = "analysis" &  
comments = "Memory for ADAMS/Car major role"
```

四轮台柱式垂向激励试验台

■ 次要角色(Minor role)

- ◆ ADAMS/Car 模板和试验台中的次要角色存储在 ADAMS/View 中的变量名称为 minor_role。试验台的次要角色通常是设置为 any。这样设置相当重要，因为你需要试验台能够与其它不同次要角色的模板相匹配。例如，一个悬架的试验台应该既可以与前/后悬架实现正确匹配，也可以与拖车类型的悬架相匹配。如果试验台的次要角色被定义为 any 的话，试验台和悬架子系统之间的信息交换器就不会出现不能正确匹配的问题。

```
variable create &  
variable_name = __acme_4PostRig.minor_role &  
string_value = "any" &  
comments = "Memory for ADAMS/Car minor role"
```

四轮台柱式垂向激励试验台

■ 模型类别 (Model class)

- ◆ 在 ADAMS/Car 中的每个模型都有一个特殊的类别。一个模型的类别存储在 ADAMS/View 下变量的名称为 model_class。此变量在模型生成的时候自动生成。当前，在 ADAMS/Car 中，共有四个模型类别，分别是：template、subsystem、test rig 和 assembly。

```
variable create &  
variable_name = .__acme_4PostRig.model_class &  
string_value = "testrig" &  
comments = "Memory for ADAMS/Car model class"
```

四轮台柱式垂向激励试验台

■ 试验台类别 (Test rig class)

- ◆ 在 ADAMS/Car 中，所有的试验台都有一个相关的特定装配类别的定义。例如，__MDI_SUSPENSION_TESTRIG 试验台与悬架装配相关。试验台的类别存储在 ADAMS/View 中变量名为testrig_class。

```
variable create &  
variable_name = __acme_4PostRig.testrig_class &  
string_value = "full_vehicle" &  
comments = "Memory for ADAMS/Car testrig class"
```

- ◆ 变量 testrig_class 直接用于图形化用户界面。例如：该变量是用选择新悬架装配还是新整车装配的对话框。这两个对话框中分别包含不同的选项，从中可以选择包含在新装配中的试验台。此可选菜单将只包含该装配类别相兼容的试验台。例如，在新悬架装配对话框中，将只列出试验台类别为悬架装配的试验台。

四轮台柱式垂向激励试验台

■ 准备生成个人二进制文件

- ◆ 现在你将增加你新的试验台模型文件 (acme_4PostRig.cmd) 和其它本练习附带的一些宏文件一起生成用户化的 ADAMS/Car 二进制文件，该文件中将包含新的试验台。生成个人和定点二进制文件的过程如下所述：
 1. 如果不存在目录，先生成一个目录 C:\acar_private (MSC.ADAMS在运行个人二进制文件时总是先搜索 \$HOME\acar_private，除非系统变量 MDI_ACAR_PRIVATE 明确的定向到其它目录)。
 2. 拷贝下面五个 ASCII 格式的文件到此目录下(C:\acar_private)：
 - acar_build.cmd
 - acme_4postrig.cmd
 - macros_ana.cmd
 - mac_ana_ful_fou_sub.cmd
 - acme_four_sim.cmd
- ◆ 下面分别介绍每个文件的功用：
 - 1、acar_build.cmd

四轮台柱式垂向激励试验台

- ◆ 此文件在建立个人二进制文件时 MSC.ADAMS 将调用，通常来讲，该文件中可以包含任何的 ADAMS/View 的命令以：
 - 修改 MSC.ADAMS/Car 的界面
 - 导入试验台模型文件
 - 增加库 (参考命令导航器 - command navigator)
 - 增加宏
 - 以及一些标准的命令
- ◆ 下面为 acar_build.cmd 文件的部分内容：

```
!---- Create custom libraries for storage ----  
library create library_name=.ACME  
library create library_name=.ACME.macros
```

- ◆ 上面两行将生成所有的 acme 宏的存储位置。

```
!---- Read analysis macros and model ----  
file command read file="C:\acar_private\macros_ana.cmd"  
file command read file="C:\acar_private\acme_4postrig.cmd"  
file command read file="C:\acar_private\acme_four_sim.cmd"
```

四轮台柱式垂向激励试验台

- ◆ 下面 acar_build.cmd 中还需要包含一些调用其它的文件，这些文件在生成用户化的二进制文件时所需要的。在此练习中，我们将调用其它三个 cmd 文件 (本练习后面将介绍)。注意这些文件的位置直接加在 acar_build.cmd 文件中。另外，该文件中还包含其它的一些标准的命令，目的是不显示某些栏目：

```
!---- Standard command to undisplay model  
model display model=(NONE)  
  
!---- Standard Command for Message Box  
interface dialog execute &  
  dialog_box_name=.gui.msg_box &  
  undisp=yes
```

- ◆ 编辑此文件以使其中的命令适合你的计算机。

2、acme_4PostRig.cmd

- ◆ 此文件包含你刚刚生成的试验台的模型文件 (.cmd)。此试验台模型将被读入你的个人二进制文件中。

3、macros_ana.cmd

四轮台柱式垂向激励试验台

- ◆ 此文件起到类似于到文件 `mac_ana_ful_fou_sub.cmd` 的指针的作用。该文件中包含一个文件绝对的位置。相比较直接读入仿真的宏的方式而言，使用这种指针的方式是比较好的。采用这种方式使得修改更容易进行。

- ◆ 编辑此文件，使之跟你的机器相一致。

4、`mac_ana_ful_fou_sub.cmd`

- ◆ 此为一个控制 ADAMS/Car 如何进行仿真模型的宏。有关该文件的深入讨论在 ADAMS/Car 的用户化分析部分讨论，见 546页。

5、`acme_four_sim.cmd`

- ◆ 此为用户化对话框的修改，其中增加了一个在菜单 Tools 下特别的 Acme Four Post Simulation 项目。该文件使用文本编辑工具生成的，并增加了一些注释。重新定制菜单的命令在该文件的最后部分。生成你自己的对话框和菜单是很容易的，但使用这种文件的方式更为有效。

四轮台柱式垂向激励试验台

- 在个人二进制文件中增加四轮台柱分析试验台
- 要生成个人二进制文件：
 1. 可以：
 - ◆ Windows: 从 **Start** —— **Programs** —— **MSC.Software** —— **MSC.ADAMS 2005** —— **A/Car** , 然后选择 **Advanced**。在提示中输入 **cr-privatebin**。
 - ◆ UNIX : 在一个 shell 窗口 , 输入 **adams05r2 -c acar cr-privatebin**。
 2. 检查一下 **acar.log** 文件 , 看看在生成 **acar.bin** 二进制文件过程中所出现的信息或错误。

四轮台柱式垂向激励试验台

■ 使用个人二进制文件运行 ADAMS/Car

要使用个人二进制文件运行 ADAMS/Car：

◆ 在 **Windows** 系统上：

- 从 **Start** —— **Programs** —— **MSC.Software** —— **MSC.ADAMS 2005r2** —— **A/Car**，然后选择 **Advanced**。在提示窗口输入 **ru-private**。

- 你也可以生成一个快捷方式，注意包含路径：

C:\Program Files\ADAMS05r2\common\mdi.bat acar ru-private.

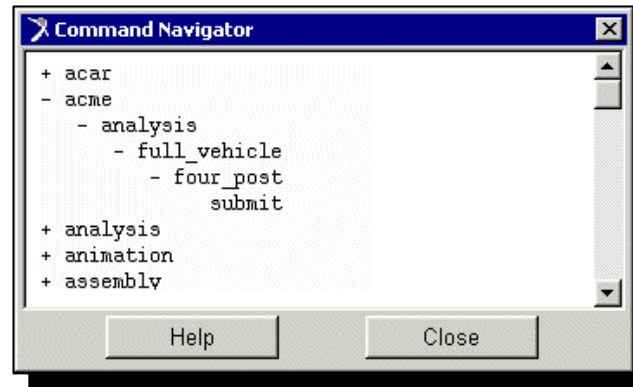
注意：当你想要随后再生成包含第二个试验台的二进制文件，你需要确认 **acar_build.cmd** 命令文件中既调用了你的新文件，也调用了原来的旧文件！

◆ 在 **UNIX** 系统上：

- 在 shell 窗口下输入 **adams05r2 -c acar ru-private**。

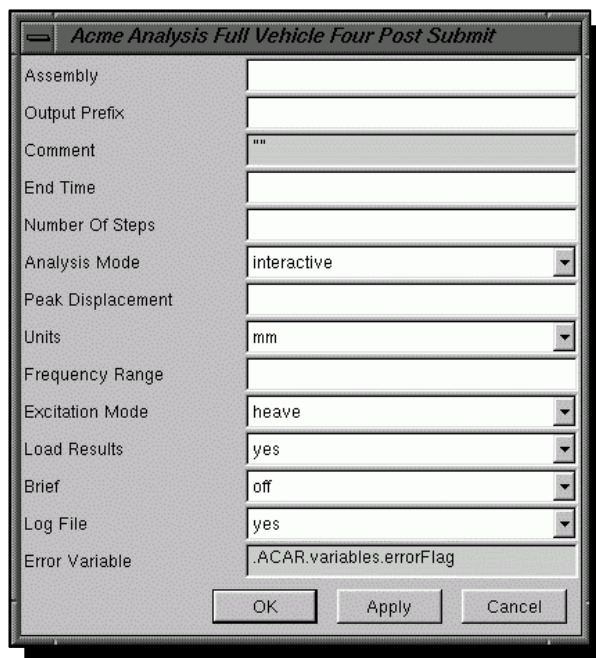
四轮台柱式垂向激励试验台

■ 界面



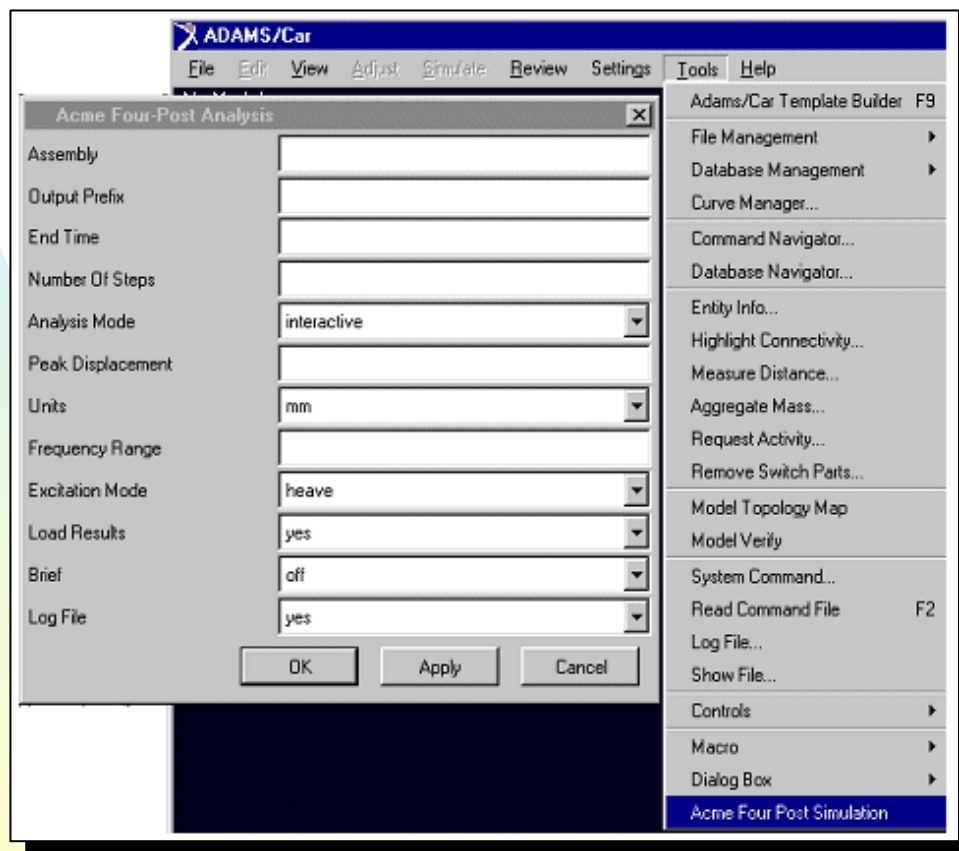
- ◆ 要测试四轮台柱分析的宏的最简单的方式就是通过 Command Navigator 直接调用该命令。要执行的命令是由宏文件 macros_ana.cmd 中的 user_entered_command 指定的。当你从 Command Navigator 中执行四轮台柱分析的宏时，ADAMS/Car自动地基于宏中的参数生成一个图形化的用户界面 (或称作对话框-dialog box)，你可以使用该对话框执行该宏并提交四轮台柱式垂向激励试验的分析：

四轮台柱式垂向激励试验台



- ◆ 要生成你自己的界面也很容易。在一个文本编辑工具中打开文件 `acme_four_sim.cmd`。这是一个用户化对话框修改，其中在菜单 Tools 下增加了一项 Acme Four Post Simulation 的条目其执行的结果如下页所示：

四轮台柱式垂向激励试验台



四轮台柱式垂向激励试验台

■ 使用该试验台生成一个装配

- ◆ 到现在，你已经建立了试验台的模板、试验台模型文件 (.cmd)、分析的宏、个人二进制文件和用户化界面。现在你可以对一个整车装配进行四轮台柱式垂向激励的分析了。

■ 要生成一个装配：

- ◆ 在 **Windows** 操作系统上：

1. 从 **Start** —— **Programs** —— **MSC.Software** —— **MSC.ADAMS 2005r2** —— **A/Car**，然后选择 **Advanced**。
2. 在提示窗口输入 **ru-private**。继续进行步骤 4。

- ◆ 在 **UNIX**操作系统上：

3. 在 shell 窗口下输入 **adams05r2 -c acar ru-private**。继续进行步骤 4。
4. 在 ADAMS/Car 提示你要使用什么模式时。确认选择 **Standard Interface**。
5. 从菜单 **File** 选择 **New** 再选择 **Full-Vehicle Assembly**。

四轮台柱式垂向激励试验台

6. 在 **Assembly Name** 文本栏内，输入 **test_4post_vehicle**。
7. 在 **Front Susp Subsystem** 栏内点击鼠标右键并选择 **Search**, 再选择 **<shared>/subsystems.tbl**，选择 **TR_Front_Suspension.sub**。
8. 在 **Rear Susp Subsystem** 栏内按照上一步的方法选择 **TR_Rear_Suspension.sub**。
9. 在 **Steering Subsystem** 栏内选择 **TR_Steering**。
10. 在 **Front Wheel Subsystem** 栏内选择 **TR_Front_Tires.sub**。
11. 在 **Rear Wheel Subsystem** 栏内选择 **TR_Rear_tires.sub**。
12. 在 **Body Subsystem** 栏内选择 **TR_Body.sub**。
13. 确认 **Powertrain Subsystem** 未选。
14. 确认 **Brake Subsystem** 未选。
15. 设置 **Vehicle Test Rig** 为 **__acme_4PostRig**。
16. 选择 **OK**。

完成装配的过程需要一些时间。

四轮台柱式垂向激励试验台

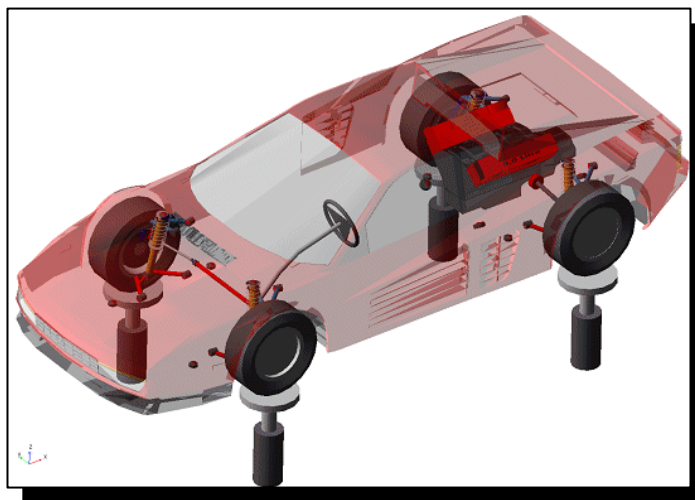
■ 要使用 Delft 轮胎模型：

1. 在车辆的左-前轮上点击鼠标右键并选择 **Wheel:whl_wheel**，然后再选择 **Modify**。
2. 修改轮胎特性 **Property File** 为 **<shared>\tires.tb\mdi_delft01.tir**。
3. 选择 **OK**。
4. 在车辆的左-后轮上点击鼠标右键并选择 **Wheel:whl_wheel**，然后再选择 **Modify**。
5. 修改轮胎特性 **Property File** 为 **<shared>\tires.tb\mdi_delft01.tir**。
6. 选择 **OK**。

四轮台柱式垂向激励试验台

■ 要失效后传动：

1. 从菜单 **View** 下选择 **Subsystem**。
 2. 选择 **TR_Rear_Suspension**，再选择 **OK**。
 3. 从菜单 **Adjust** 下选择 **Driveline Activity**。
 4. 设置 **Current Mode** 为 **Inactive**，再选择 **OK**。
 5. 从菜单 **View** 下选择 **Assembly**，再选择 **OK**，这样显示整个装配。
- 注意到车辆右侧的轮胎自动地进行了调整，这是因为几何对称性的缘故。显示的模型如下：



四轮台柱式垂向激励试验台

运行四轮台柱式垂向激励仿真

要运行一次一般的四轮台柱式垂向激励仿真：

1. 从菜单 **Tools** 下选择 **Acme Four-Post Analysis**。
2. 按照下图填入参数：

Parameter	Value
Assembly	.test_4post_vehicle
Output Prefix	FourPost_test
End Time	2
Number Of Steps	600
Analysis Mode	interactive
Peak Displacement	150
Units	mm
Frequency Range	3
Excitation Mode	heave
Load Results	yes
Brief	off
Log File	yes

3. 选择 **OK**。

四轮台柱式垂向激励试验台

- 仿真过程要持续一段时间。
- 要观察动画：
 1. 从菜单 **Review** 下选择 **Animation Controls**。
 2. 选择 **Play** 。

你也可以在 ADAMS/PostProcessor 观察仿真结果的曲线，退出 ADAMS/Car，选择 Exit。

现在存储装配并退出。
- 要存储装配并退出：
 1. 从菜单 **File** 下选择 **Save** 再选择 **Assembly**。
 2. 选择 **Close** assembly after save。
 3. 选择 **OK**。

ADAMS/Car 存储该装配在你个人的数据库下。

四轮台柱式垂向激励试验台

■ 修改整车装配

■ 打开刚刚生成的装配并检查一下试验台是如何指定的：

1. 在 ADAMS/Car 外部，找到你的个人数据库目录并打开装配目录。
2. 在一个文本编辑工具中打开文件 **test_4post_vehicle.asy**。
3. 向下滚动直至找到下面的内容：

```
$-----TESTRIG  
[TESTRIG]  
USAGE = "__acme_4PostRig"
```

4. 关闭该文件。

注意：如果你现在就有一个与试验台 __MDI_SDI_TESTRIG的装配，你只要编辑该装配文件，并重新定义试验台__acme_4PostRig即可。

四轮台柱式垂向激励试验台

- 在 ADAMS/Car 中增加用户化的分析过程
 - ◆ 要在 ADAMS/Car 中增加用户化的分析过程的基本前提是用户对 ADAMS/View 的宏语言的通透的理解。在这一节练习过程中，你可以参考 ADAMS/Car 在线帮助中 Customize 部分。
 - ◆ ADAMS/Car 本身是一种开放式的架构，很容易基于用户的需求进行功能上的扩展。用户使用 ADAMS/Car 时，大多数情况下，是扩展其分析功能，以包括一些其公司内部的分析类型或操纵分析，而这些分析并不包括在标准的 ADAMS/Car 产品下。在 ADAMS/Car 所有的分析都是使用宏(macros)的方式进行的。因此，在 ADAMS/Car 主增加一个新的分析可能只是简单的增加一个新的宏而已。一个使用该宏的用户化的图形界面也可以生成以方便使用，但不是必须的。
 - ◆ 本例中将提供你一步步的指导以生成你自己的宏。分析的目的在 Four-Post Vertical Excitation Test 文档中详细描述。

四轮台柱式垂向激励试验台

■ 开始练习

- ◆ 要开始生成新的分析的最佳途径并不是从草稿直接开始建立。在 ADAMS/Car 产品自带的可能进行的分析类型很广泛。查看一下在现有的功能中哪一个分析最接近你需要进行的分析。一旦你选择了一个现有的宏，只是需要简单地修改一下该宏以使其适合你的需求。
- ◆ 你能够使用 ADAMS/Car 中内置的宏编辑器观察任何标准的宏。从 ADAMS/Car 菜单下选择 Tools -> Macro -> Edit -> Modify 以打开数据库导航器 (Database Navigator)。然后选择你想要显示的宏。举例来说，从数据库导航器 (Database Navigator) 中选择 mac_ana_ful_fou_sub，允许你观察用于递交整车分析的标准的宏中的内容。



更多有关 ADAMS/Car 中宏的信息，参见 ADAMS/Car 在线帮助中有关 Customize 部分。

四轮台柱式垂向激励试验台

- ◆ 只要可能，在用户化分析的宏中应该使用一个退出宏。使用这样一个 ADAMS/Car 中自带的退出宏有以下好处：其一，这些宏是由 MSC.Software 所写并经过测试的，应该不会有问题；其二，使用这些宏可以减少在 ADAMS/Car 中增加用户化分析的工作量；其三，这些宏提供了进行用户化开放的公共平台，维护起来比较方便，在不同的项目直接进行调试也比较方便。

■ 分析宏的架构

- ◆ 通常情况下，在 ADAMS/Car 中所有的分析宏的架构都是一样的。特别是对于分析类的宏更是如此。比如，所有的开环整车分析宏都是极其相似的。所有的整车分析宏都是基于驾驶机器 (Driving Machine)。典型的，宏中只有一小部分用来使得宏和分析的特别之处。在 ADAMS/Car 中每一个分析宏都有相同的基本架构：

四轮台柱式垂向激励试验台

参数定义

- ◆ 参数为最终用户可以在执行宏时输入的参数。在执行该宏的时候需要提供这些参数的值。



宏的参数在 ADAMS/View 的在线帮助中有详尽的描述。

- ◆ 在 ADAMS/Car 所有的分析宏中都包含两个通用的参数，这两个参数为：assembly 和 output_prefix 参数。Assembly 参数用于指定在该分析过程中 ADAMS/Car 要分析哪个装配模型；而 output_prefix 参数则用来指定在进行分析时所生成的输出文件要增加的前缀。这些参数通常在分析宏中最先定义的，如下面的例子所示：

四轮台柱式垂向激励试验台

- ◆ 四轮台柱分析的参数由附带的 Four-Post Vertical Excitation Test 的文档所确定。包含对每个参数的值对于成功进行四轮台柱分析的重要性的描述。

```
! $assembly:t=model  
! $output_prefix:t=string  
! $comment:t=string:d= " "  
! $end_time:t=real:gt=0  
! $number_of_steps:t=integer:gt=0  
! $analysis_mode:t=list(interactive,graphical,background):d=interactive  
! $peak_displacement:t=real:gt=0  
! $units:t=list(mm):d=mm  
! $frequency_range:t=real:gt=0  
! $excitation_mode:t=list(heave,pitch,roll,warp):d=heave  
! $load_results:t=list(yes,no):u=yes  
! $brief:t=list(on,off):u=off  
! $log_file:t=list(yes,no):u=yes  
! $error_variable:t=variable:d=.ACAR.variables.errorFlag
```

四轮台柱式垂向激励试验台

■ 参数描述：

- ◆ Assembly：该参数需要用户指定一个已经打开了模型。
- ◆ analysis_name：一个字符串，代表此分析所有输出文件名字的前缀。
- ◆ end_time：一个实数，ADAMS/Solver 进行四轮台柱分析结束时间。
- ◆ number_of_steps：一个整型数，ADAMS/Solver 进行分析输出的步数。
- ◆ analysis_mode：一个字符串，指定分析的模式。有两个正确的模式可选：interactive 或 background。
- ◆ peak_displacement：激振试验台每个立柱最大的垂向位移量。
- ◆ Units：该练习中直接内置为 mm，可以扩展包含其它单位。
- ◆ frequency_range：一个实数，表示激励驱动的频率范围。
- ◆ excitation_mode：列表值，表示激振台柱运动的方向。
- ◆ log_file：表示是否生成分析过程的记录文件。

四轮台柱式垂向激励试验台

容错处理

- ◆ 每个分析宏中的容错处理部分应该包含检查：不正确的装配或用户输入的数值不合适。容错处理部分还应该包含检查：保证装配中包含进行要做的分析所需要的子系统。例如，如果你生成一个分析宏用来执行整车的直线制动工况的仿真，需要进行装配中是否包含制动子系统的检查。容错处理部分也应该检查用户输入的参数是否合理。
- ◆ 四轮台柱分析需要使用试验台 `__acme_4PostRig` 执行。装配和试验台的设置，将在后面讲述。另外，还要检验是否使用了正确的试验台，还要确定分析所使用的名字是否是唯一的。

四轮台柱式垂向激励试验台

```
variable set variable_name=$error_variable integer=0

!---- Check to ensure the assembly has the proper testrig ----
if condition=($assembly.original_testrig_name != "__acme_4PostRig")
    acar toolkit warning &
    w="Analysis cannot be submitted!", &
    "The assembly does not have the proper testrig. This analysis only", &
    "works with assemblies using the '__acme_4PostRig' testrig."
    variable set variable_name=$error_variable integer=1
    return
end

!---- Check if analysis name already exists ----
if condition=(db_exists("$assembly.$"output_prefix"_fourpost"))
    if condition=(alert("question","An analysis called
\\$'output_prefix'_fourpost\\ already exists. Overwrite it?","Yes","No","",2) == 2)
        variable set variable_name=$error_variable integer=1
        return
    end
end
end
```

- ◆ 用于上面例子中的 acar toolkit warning 命令将在一个小窗口中显示用户信息，该窗口将一直打开，直至你选择 OK 为止。这个功能主要用于提示用户输入错误。

四轮台柱式垂向激励试验台

■ 读入特性文件

- ◆ 在 ADAMS/Car 中，很多力元件都是从特性文件中读入其非线性特性的。在一个分析执行之前，此信息必须进行更新。如果特性文件不重新读入的话，力元件可能会包含不正确的信息，并直接影响仿真结果的准确性。
- ◆ 在验证了装配的正确性之后，特性文件信息需要必须再次读进来并进行相应的赋值。下面的例子可以直接用于用户所写的分析宏中。如前面所述，\$assembly 参数必须为一个 ADAMS/Car 的装配。特性文件通过用户函数，如下所述，读进来。下面的例子主要是演示如何使用 acar toolkit message 这个可利用的宏输出文本信息到信息窗口中。在分析递交到 ADAMS/Solver 之前，读入特性文件非常重要。

四轮台柱式垂向激励试验台

```
!---- Clear out message window ----  
acar toolkit message &  
message=" " append=no display=no closeable=yes echo_to_logfile=no  
  
!---- Read property files ----  
acar toolkit message &  
message="Reading in property files..."  
variable set variable_name=$_self.readEm &  
  
integer_value=(eval(read_property_file($assembly)))  
acar toolkit message &  
message="Reading of property files completed."
```

■ 装配和试验台的设置

- ◆ 装配和试验台的设置是分析宏中的一部分，也是该分析宏区别于其它宏的部分。在这一部分，基于所作的分析类型和用户所指定的参数修改试验台和装配。

四轮台柱式垂向激励试验台

- ◆ 四轮台柱试验分析的代码部分如下所示，包括一个开始部分的描述：

```
!---- Setup the assembly for the maneuver ----  
acar toolkit message &  
message="Setting up vehicle assembly for four post shaker..."
```

- ◆ 轮胎载荷的参考标记点需要重新分配到试验台相应合适的台柱上。参考标记点的信息交换器变量的命名规则是按照固定的方式考虑的，在宏中搜索四轮台柱试验台中已知的信息交换器。注意，对于一个特定的装配，轮胎参考标记点的设置只进行一次。如果同样的装配用于多次四轮台柱的分析，初始的设置对每次分析都是正确的。

四轮台柱式垂向激励试验台

- ◆ 对于每个车轮，轮胎参考标记点都被分配到相应的台柱上。第一步就是首先找到整车装配中的所有的车轮。通过试验台的输出信息交换器再进行重新分配。



有关信息交换器的信息，参见 ADAMS/Car 在线帮助中 Templates 部分。

- ◆ 信息交换器中保留了激振台柱的名称，轮胎载荷的参考标记点应该属于对应的台柱部件。

```
if condition=(!db_exists("$assembly.fourpostSetup"))
!---- Parameterize the 4post pad height to the global road height marker just previously adjusted ----
variable set variable=$_self.frontWheel &
object_value=(eval(subsystem_lookup($assembly,"wheel","front")))
variable set variable=$_self.leftFrontWheel &
object_value=(eval(db_filter_name(db_children($_self.frontWheel[1],"ac_tire"),"til_*"))
variable set variable=$_self.rightFrontWheel &
object_value=(eval(db_filter_name(db_children($_self.frontWheel[1],"ac_tire"),"tir_*"))
variable set variable=$_self.rearWheel &
object_value=(eval(subsystem_lookup($assembly,"wheel","rear")))
variable set variable=$_self.leftRearWheel &
object_value=(eval(db_filter_name(db_children($_self.rearWheel[1],"ac_tire"),"til_*"))
variable set variable=$_self.rightRearWheel &
object_value=(eval(db_filter_name(db_children($_self.rearWheel[1],"ac_tire"),"tir_*"))
marker modify &
marker_name=(eval($_self.leftFrontWheel.object_value.ref_marker.object_value)) &
new_marker_name=(eval($assembly.testrig.col_front_pad_mount[1]/"/"."//
$_self.leftFrontWheel.object_value.ref_marker.object_value.name))
```

四轮台柱式垂向激励试验台

```
marker modify &
marker_name=(eval($_self.rightFrontWheel.object_value.ref_marker.object_value)) &
new_marker_name=(eval($assembly.testrig.cor_front_pad_mount[1]//". "//
$_self.rightFrontWheel.object_value.ref_marker.object_value.name))
marker modify &
marker_name=(eval($_self.leftRearWheel.object_value.ref_marker.object_value)) &
new_marker_name=(eval($assembly.testrig.col_rear_pad_mount[1]//". "//
$_self.leftRearWheel.object_value.ref_marker.object_value.name))
marker modify &
marker_name=(eval($_self.rightRearWheel.object_value.ref_marker.object_value)) &
new_marker_name=(eval($assembly.testrig.cor_rear_pad_mount[1]//". "//
$_self.rightRearWheel.object_value.ref_marker.object_value.name))
variable set variable=$assembly.fourpostSetup &
integer_value=1
end
```

四轮台柱式垂向激励试验台

- ◆ 对每次单独的四轮台柱分析，运动激励所驱动在台柱的位移量都应该重新设置。其位移量是相对于前面所讲的轮胎的参考坐标系标记点，对特定的装配只需要再设置一次，并对所有的成功的四轮台柱分析保持正确。四种激励模式为：heave、pitch、roll和warp。每种激励模式的振动幅值是一样的，但运动的方式方向是由特定的激励方式决定的。在heave模式下，所有的四轮台柱都将同方向运动。在pitch模式下，前/后轮胎将按照相反的方向运动。对于roll模式，左/右轮胎将按照相反的方向运动。而当warp模式被指定时，左前轮和右后轮与右前轮和左后轮将按照相反的方向运动。不同的激励模式是通过在驱动函数的定义中指定“1”或“-1”乘子的方式实现的。

四轮台柱式垂向激励试验台

```
!---- Assign actuator functions based on excitation mode ----
!-Heave Excitation
if condition=("$excitation_mode" == "heave")
    acar template_builder actuator set function &
    actuator=$assembly.testrig.jms_left_front_actuator &

function="1*$peak_displacement*sin(.5*360d*$frequency_range/$end_time*time**2)"
    acar template_builder actuator set function &
    actuator=$assembly.testrig.jms_right_front_actuator &

function="1*$peak_displacement*sin(.5*360d*$frequency_range/$end_time*time**2)"
    acar template_builder actuator set function &
    actuator=$assembly.testrig.jms_left_rear_actuator &

function="1*$peak_displacement*sin(.5*360d*$frequency_range/$end_time*time**2)"
    acar template_builder actuator set function &
    actuator=$assembly.testrig.jms_right_rear_actuator &

function="1*$peak_displacement*sin(.5*360d*$frequency_range/$end_time*time**2)"
!- Pitch Excitation
    elseif condition=("$excitation_mode" == "pitch")
        acar template_builder actuator set function &
        actuator=$assembly.testrig.jms_left_front_actuator &

function="1*$peak_displacement*sin(.5*360d*$frequency_range/$end_time*time**2)"
        acar template_builder actuator set function &
        actuator=$assembly.testrig.jms_right_front_actuator &

function="1*$peak_displacement*sin(.5*360d*$frequency_range/$end_time*time**2)"
        acar template_builder actuator set function &
        actuator=$assembly.testrig.jms_left_rear_actuator &

function="-1*$peak_displacement*sin(.5*360d*$frequency_range/$end_time*time**2)"
        acar template_builder actuator set function &
        actuator=$assembly.testrig.jms_right_rear_actuator &

function="-1*$peak_displacement*sin(.5*360d*$frequency_range/$end_time*time**2)"
!- Roll Excitation
    elseif condition=("$excitation_mode" == "roll")
        acar template_builder actuator set function &
        actuator=$assembly.testrig.jms_left_front_actuator &

function="1*$peak_displacement*sin(.5*360d*$frequency_range/$end_time*time**2)"
        acar template_builder actuator set function &
        actuator=$assembly.testrig.jms_right_front_actuator &
```

四轮台柱式垂向激励试验台

```
function="-1*$peak_displacement*sin(.5*360d*$frequency_range/$end_time*time**2)"
    acar template_builder actuator set function &
    actuator=$assembly.testrig.jms_left_rear_actuator &

function="1*$peak_displacement*sin(.5*360d*$frequency_range/$end_time*time**2)"
    acar template_builder actuator set function &
    actuator=$assembly.testrig.jms_right_rear_actuator &

function="-1*$peak_displacement*sin(.5*360d*$frequency_range/$end_time*time**2)"
    !- Warp Excitation
    elseif condition=("$excitation_mode"=="warp")
        acar template_builder actuator set function &
        actuator=$assembly.testrig.jms_left_front_actuator &

function="1*$peak_displacement*sin(.5*360d*$frequency_range/$end_time*time**2)"
    acar template_builder actuator set function &
    actuator=$assembly.testrig.jms_right_front_actuator &

function="-1*$peak_displacement*sin(.5*360d*$frequency_range/$end_time*time**2)"
    acar template_builder actuator set function &
    actuator=$assembly.testrig.jms_left_rear_actuator &

function="-1*$peak_displacement*sin(.5*360d*$frequency_range/$end_time*time**2)"
    acar template_builder actuator set function &
    actuator=$assembly.testrig.jms_right_rear_actuator &

function="1*$peak_displacement*sin(.5*360d*$frequency_range/$end_time*time**2)"
    end
```

四轮台柱式垂向激励试验台

■ 递交分析

- ◆ 递交一个分析到 ADAMS/Solver 的前提是必须先有个 MSC.ADAMS 的模型文件 (.adm) 和一个 MSC.ADAMS 的命令文件 (.acf)。在 ADAMS/Car 中，两个基本的分析是悬架的分析和整车的分析。在两种情况下，一个实用的宏可以用来生成 ADM 文件和 ACF 文件并直接提交 ADAMS/Solver 进行分析。这些实用的宏可以在每一个悬架或整车的分析宏中执行。
- ◆ 设置装配后，就可以进行四轮台柱的分析了。既然这是一个整车的装配，就需要使用相应的整车递交分析的实用的宏。在这种情形下，两个附加的参数需要指定，因为对于四轮台柱试验台，这两个参数没有指定：`generate_road_geometry` 和 `simulation_type`。

四轮台柱式垂向激励试验台

- ◆ 调用四轮台柱整车试验台中递交分析的宏的代码如下所示，包括重要的相关参数的描述如下：

```
!---- Perform the analysis ----  
acme analysis full_vehicle four_post submit &  
  assembly=$assembly &  
  analysis_name="$'output_prefix'_fourpost" &  
  end_time=$end_time &  
  number_of_steps=$number_of_steps &  
  analysis_mode=$analysis_mode &  
  load_results=$load_results &  
  brief=$brief &  
  road_data_file="BEDPLATE" &  
  generate_road_geometry=no &  
  simulation_type=fourpost
```

- ◆ 参数描述：
 - Assembly：该参数需要用户指定一个已经打开了模型。
 - analysis_name：一个字符串，代表此分析所有输出文件名字的前缀。

四轮台柱式垂向激励试验台

- end_time：一个实数，ADAMS/Solver 进行四轮台柱分析结束时间。
- number_of_steps：一个整型数，ADAMS/Solver 进行分析输出的步数。
- analysis_mode：一个字符串，指定分析的模式。有两个正确的模式可选：interactive 或 background。
- load_results：一个字符串，指定在分析结束时是否读入分析的结果，预计的值有两个，yes 或 no。
- road_data_file: 内置为 BEDPLATE，表示车辆并不是在路面上驾驶。ADAMS/Car 将内部的阐释并处理其内置值。
- generate_road_geometry：设置为“no”，表示 ADAMS/Car 将不基于 road_data_file 生成路面的几何外形。
- simulation_type：内置为 fourpost，表示整车将被置于四轮台柱激励试验台上。ADAMS/Car 将内部的阐释并处理其内置值以生成正确的 .adm 文件和 .acf 文件。

四轮台柱式垂向激励试验台

■ 记录分析过程

- ◆ 很多用户认为在执行分析的过程中生成记录文件非常重要。生成记录文件是非常重要的，因为该文件中存储了分析过程中的历史进程数据。存储的数据是有用的，有时需要允许用户再生成特定的分析的结果。这些实用的宏在每一个悬架或整车分析的宏中执行。
- ◆ 现有的实用的宏可以在你的用户化的分析宏中使用以生成记录文件。当分析完成，结果可能被记录到一个文件中。另外，最终的诊断信息可以显示在信息窗口中。

四轮台柱式垂向激励试验台

```
if condition=($log_file)
  acar analysis full_vehicle log &
  assembly=$assembly &
  analysis_name="$'output_prefix'_fourpost" &
  analysis_type="Four Post Shaker Test" &
  analysis_log_file="$'output_prefix'_fourpost.log" &
  comment=$comment &
  end_time=$end_time &
  number_of_steps=$number_of_steps &
  road_data_file="BEDPLATE" &
  initial_velocity=0.0 &
  velocity_units="m_sec" &
  input_parameters="general input" &
  parameter_values=("$number_of_steps")
end

if condition=("$analysis_mode" != "background")
  acar toolkit message &
  message="Simulation is complete."
end
```

■ 结尾

- ◆ 最后，保证所有的在该宏中通过 `$_self` 术语定义的局部变量被删除，这一点也是很重要的。

```
variable delete variable_name=(eval(db_children($_self,"variable"))
```

附录 C ADAMS/CAR 中的文件

- 此附录介绍 ADAMS/Car 中的配置文件和数据文件的一些信息。

ADAMS/CAR 中的文件

- 本附录内容：
 - ◆ ADAMS/Car 配置文件
 - ◆ ADAMS/Car 数据文件

ADAMS/CAR 中的文件

Table 2. ADAMS/Car Configuration Files

The file:	Contains:
.acar.cfg	Information that ADAMS/Car reads during startup to correctly initialize the session. There are private, shared, and site configuration files.
acar.cmd	Commands for starting ADAMS/Car.
acarAS.cmd	Preferences you set. AS stands for <i>After Startup</i> , meaning that ADAMS/Car reads it after it reads other setup files.
acarBS.cmd	Preferences you set. BS stands for <i>Before Startup</i> , meaning that ADAMS/Car reads it before it reads other setup files.

ADAMS/CAR 中的文件

Table 3. ADAMS/Car Data Files

The file:	Does the following:
Aero_force(.aer)	Contains wind-force mappings.
Assembly(.asy)	Lists the subsystems that make up ADAMS/Car assemblies.
Autoflex input(.afi)	Describes a section+centerline+attachment points (an extruded solid). The executable, afi2mnf.exe, processes the file to create a modal neutral file (MNF) flexible body.
ADAMS/Car database(.cdb)	Directory that serves as the ADAMS/Car database.
Driver control(.dcf)	Contains maneuver descriptions for the Driving Machine.
Driver data(.dcd)	Contains data for the Driving Machine.
Differential(.dif)	Defines the slip speed-torque characteristics of a differential.
Driver loadcase(.dri)	Contains driving signals used in a data-driven, full-vehicle analysis. The driver loadcase specifies inputs to the vehicle.
Loadcase(.lcf)	Contains data used in suspension analyses.

ADAMS/CAR 中的文件

Table 3. ADAMS/Car Data Files (continued)

The file:	Does the following:
Model (.mdl)	Obsolete
Plot configuration (.plt)	Defines a suite of plots to be automatically generated after completion of an analysis.
Powertrain (.pwr)	Defines the engine speed-torque relationship at different throttle positions.
Property	Contains force properties for the entities: <ul style="list-style-type: none">u Bump stops (.bum), .xmlu Bushings (.bus), .xmlu Dampers (.dpr), .xmlu Reboundstop (.reb), .xmlu Spring (.spr), .xmlu Tire (.tir)
Road Data (.rdf)	Contains data on road.
Subsystem (.sub)	Contains information unique to the specific instance of the template the subsystem file references.
Suspension curves (.scf)	Used in the Conceptual Suspension Modeling add-on module.
Steering_assist (.ste)	Contains torsion bar data relating torsion bar deflection to both torque and pressure.
Tables (.tbl)	Subdirectory in the ADAMS/Car database called tables. Each subdirectory contains files for specific types of components, such as springs and dampers, or files for performing tests, such as loadcases and wheel envelopes.
Template file (.tpl)	Defines the topology and major role (for example, suspension or steering) of ADAMS/Car models.
Wheel envelope (.wen)	Contains location vector information that represents the wheel center location and orientation in space. Used for wheel envelope analyses.